



TREBALL FINAL DE GRAU



ESCOLA
POLITÈCNICA SUPERIOR
UNIVERSITAT DE LLEIDA
INSPIRING THE FUTURE

Estudiant: Caminero Lozano, Ricardo

Titulació: Grau en Enginyeria Electrònica Industrial i Automàtica

Títol de Treball Final de Grau: **DISEÑO DE UN CANCELADOR DE INTERFERENCIAS. IMPLEMENTACIÓN EN MICROCONTROLADOR CON FILTROS ADAPTATIVOS.**

Director/a: Clariá Sancho, Francisco

Presentació

Mes: Juny

Any: 2018

Tabla de contenido

1	Introducción.....	3
1.1	Objetivos.....	4
2	Filtros adaptativos.....	5
2.1	Ruido Aleatorio	5
2.2	Funcionamiento	6
2.3	Filtros lineales.....	8
2.3.1	Filtro de respuesta infinita al impulso (IIR).....	9
2.3.2	Filtro de respuesta finita al impulso (FIR).....	10
2.3.3	Filtro adaptativo basado en el combinador lineal	11
2.3.4	Filtro adaptativo de error cuadrático medio mínimo (LMS)	12
2.4	Aplicaciones.....	14
2.4.1	Identificador.....	14
2.4.2	Predictor	14
2.4.3	Ecualizador-Modelado inverso	15
2.4.4	Cancelador de interferencias	15
2.5	Conclusiones	17
2.6	Bibliografía	18
3	Algoritmo LMS.....	20
3.1	Introducción	20
3.2	Estructura del algoritmo LMS.....	20
3.3	Conclusiones	22
3.4	Bibliografía	22
4	Experimento práctico	24
4.1	Elementos necesarios.....	24
4.1.1	Microcontrolador	24
4.1.2	Acelerómetros.....	26
4.1.3	Motores de corriente continua	26
4.1.4	Atollic TrueSTUDIO	27
4.1.5	Tera Term.....	27
4.1.6	Matlab.....	27
4.2	Simulación del experimento	28
4.3	Realización del experimento	31

4.3.1	Obtención de la planta	31
4.3.2	Microcontrolador como cancelador de interferencias	41
4.3.3	Simulación en <i>Matlab</i>	43
4.4	Bibliografía	45
5	Conclusiones	46
5.1	Coste del estudio	47
5.1.1	Elementos del proyecto	47

1 Introducción

Desde hace tiempo existe la necesidad de modelizar matemáticamente una gran cantidad de sistemas y de crear controladores para conseguir un comportamiento deseado. Estas modelizaciones tienen un amplio rango de aplicación en áreas como la robótica, domótica, industria, medicina, aeronáutica, aplicaciones militares, etc.

El primer análisis formal de un sistema de control y desde el cual, se considera el inicio del campo de la teoría de control, fue desarrollado por James Watt en 1788, que estudio el comportamiento del regulador centrífugo y lo aplico a una caldera de vapor para ajustar la válvula de admisión. Desde entonces los controladores han ido evolucionando, separándose en dos tipos, controladores de lazo abierto y controladores de lazo cerrado, la teoría de control se centra en estos últimos.

A partir de 1955 da comienzo la Teoría Moderna de Control y empiezan a desarrollarse los primeros sistemas de control adaptativo, que tienen una importancia notable en la aeronáutica. El gran impulso de los controles adaptativos llega con la introducción de las computadoras modernas que permiten la implantación de avanzados algoritmos de control.

Con los controladores adaptativos se abre una nueva ventana en el procesado digital de señales, que no tenía demasiada incidencia en los controladores clásicos, unido al hecho que actualmente cada vez toman más importancia los sistemas digitales, en especial los dispositivos electrónicos microcontroladores. Se ha decidido en este proyecto combinar ambos temas y desarrollar un filtro adaptativo que sea capaz de cancelar las interferencias que se hallen inmersas en una señal e implementarlo en un microcontrolador. De esta forma se estudiará a nivel teórico y de forma simulada la implementación del filtro utilizando la herramienta de software matemático *Matlab* y a nivel práctico usando una placa de desarrollo *STM32F4DISCOVERY* y acelerómetros que proveerán las señales necesarias.

Durante el proyecto veremos que es un filtro adaptativo, el algoritmo de mínimos cuadrados o LMS para encontrar los coeficientes del filtro que nos permitirán obtener el valor esperado mínimo del cuadrado de la señal de error, la simulación en *Matlab* y por último su implementación en el microcontrolador.

1.1 Objetivos

Partiendo de la introducción anterior podemos definir unos objetivos claros para este proyecto.

- Objetivo principal
 - Diseñar un cancelador de interferencias que sea capaz de operar implementado en un microcontrolador. Debe procesar una señal interferida y obtener la señal deseada, eliminando la interferencia sin perder información durante el proceso.
 - Realizar un proceso experimental controlado que nos permita demostrar que el filtro diseñado es eficaz y consigue un filtrado óptimo de una señal interferente que tiene el mismo rango frecuencial que la señal de medida interferida.
 - Comparativa entre el experimento simulado y el experimento a tiempo real.
- Objetivos secundarios
 - Estudiar, analizar y escoger el tipo de filtro adaptativo que se utilizará y sus aplicaciones.
 - Estudiar, analizar y escoger el tipo de algoritmo en el que se basará el filtro adaptativo.

2 Filtros adaptativos

Antes de ver que es un filtro adaptativo es necesario saber que es un filtro y porqué ha sido necesario su uso y su invención.

Un filtro es básicamente, un elemento que elimina un cierto número de frecuencias o que las realza, cuando la señal pasa a través de él, además de poder modificar su amplitud y su fase.

Los primeros filtros que aparecieron se fabricaban usando resistencias, condensadores y bobinas, también llamados filtros pasivos. Filtros activos son aquellos que incorporan, a los elementos anteriores, transistores o circuitos integrados. En nuestro caso optaremos por usar los filtros adaptativos que pueden catalogarse como filtros activos. A continuación, se muestra un esquema de los tipos de filtros.



Figura 1. Tipos de filtros

Un filtro adaptativo es un tipo de filtro, cuyos parámetros internos, varían en el tiempo para adaptarse a la señal. En la actualidad se implementa con elementos activos y de forma digital.

2.1 Ruido Aleatorio

La función principal de los filtros es permitirnos extraer la información que consideramos útil de una señal temporal, atenuando, amplificando y eliminando sus componentes de frecuencia con las que está formada.

El ruido es una señal no deseada, añadida a otra de interés. El ruido puede originarse de muchas formas distintas, puede ser óptico,

magnético, térmico etc. Y se suele definir como cualquier señal anómala o perturbación que provoca que la información de interés quede degradada.

Cuando se trata una señal como la que trataremos en este proyecto, es decir una señal eléctrica, independientemente de que sea analógica o digital, puede verse afectada por ruidos generados a partir de la propia señal, es decir que es ruido correlacionado, y por ruido no correlacionados, que no son generados a partir de la propia señal. A pesar de que no trataremos en profundidad el tema del ruido podemos ver a continuación un esquema de los tipos de ruidos a los que se puede ver expuesta una señal eléctrica.

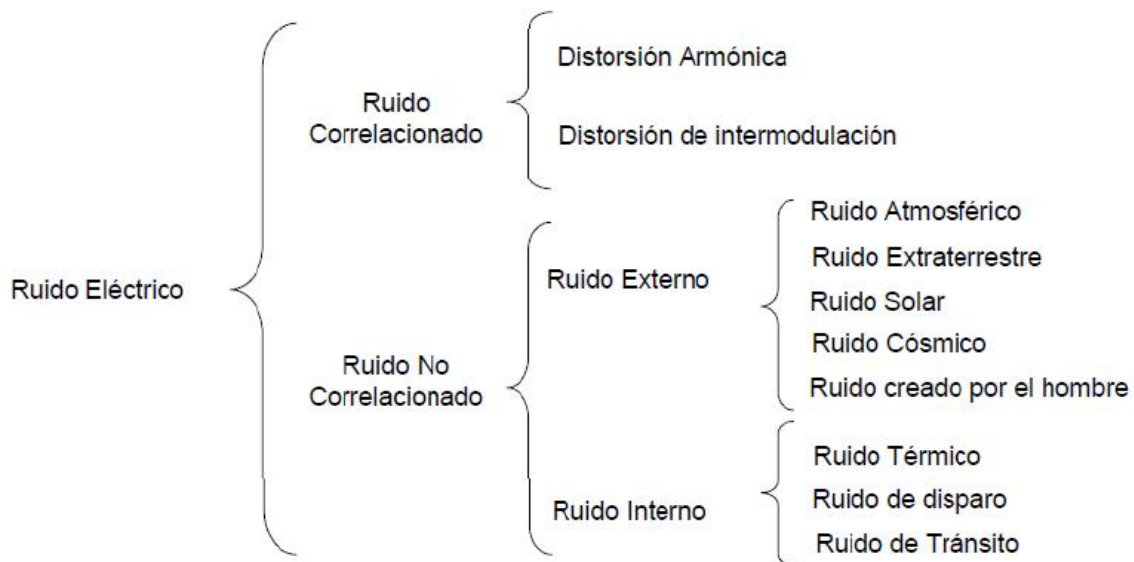


Figura 2 Tipos de ruidos eléctricos

No hay una metodología estándar para cancelar todos los tipos de ruido que contaminan una señal. Por su propia naturaleza la señal ruidosa es impredecible y cuando ocupa la misma de frecuencias que la señal que contamina no es posible cancelar este ruido, sin considerar alguna característica específica del tipo de ruido contaminante. Esto da lugar a la aplicación de un filtrado adaptativo que saca partido del conocimiento que se tiene de estas características particulares del ruido o de la interferencia.

2.2 Funcionamiento

Como ya hemos visto antes un filtro adaptativo trata de cambiar su respuesta y se ajusta para conseguir un resultado deseado. A diferencia de los filtros Wiener que requieren información previa sobre los datos que

debe procesar para poder diseñarlos óptimamente, los filtros adaptativos usan un algoritmo de aprendizaje que se basa en un método de prueba y error y se optimiza en cada iteración. Por otra parte, su implementación de forma analógica requiere circuitos electrónicos bastante complejos, sin embargo, llevarlo a cabo de forma computacional mediante un microcontrolador nos permite simplificar y abaratar su coste.

El algoritmo utilizado siempre se alimenta de una función que marca cual es el criterio para el funcionamiento óptimo del filtro, la cual determina como deben cambiar ciertos parámetros de su función de transferencia para aproximar la señal de salida al valor deseado en la próxima iteración. El algoritmo que más se suele utilizar es el LMS (*Least Mean Square*) el cual veremos más adelante.

La estructura básica de un sistema de filtro adaptativo es la siguiente,

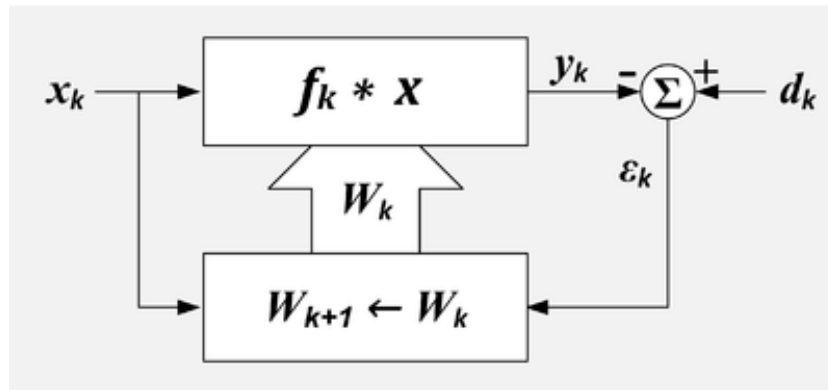


Figura 3 Diagrama de bloques filtro adaptativo

Donde:

- k = Número de muestras.
- x = Entrada de referencia
- X = Conjunto de valores recientes de x
- d = Entrada deseada
- W = Conjunto de coeficientes del filtro
- ε = Error de salida
- f = respuesta del filtro al impulso
- $*$ = convolución
- Caja superior=filtro lineal
- Caja inferior=Algoritmo adaptativo

De forma más simplificada

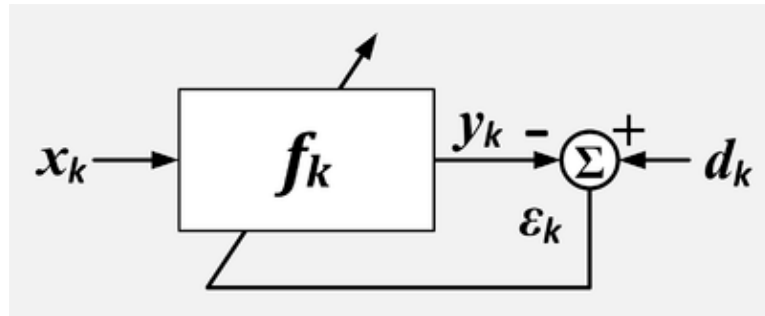


Figura 4 Diagrama de bloques simplificado filtro adaptativo

Existen tres factores principales para definir las prestaciones de un algoritmo adaptativo:

- *Complejidad computacional*: definida como el número de operaciones que han de realizarse en cada instante para implementar el algoritmo adaptativo y el tamaño de la memoria disponible para guardar los datos y el programa.
- *Velocidad de convergencia*: Esto se define como el número de iteraciones requeridas por el algoritmo para alcanzar la solución óptima. Desafortunadamente, la velocidad de convergencia es inversamente proporcional a la complejidad.
- *Desajuste*: Desafortunadamente, la velocidad de convergencia es inversamente proporcional a la complejidad.

Otros factores son:

- *Rastreo*: La facilidad para detectar las variaciones en la señal.
- *Robustez*: La capacidad para evitar grandes errores de estimación.
- *Estructura*: La optimización de su programación.
- *Propiedades numéricas*: La acumulación de errores de cálculo.

2.3 Filtros lineales

Los filtros lineales procesan señales que varían en el tiempo para producir señales de salida. Se llaman filtros lineales porque las señales de entrada y de salida de estos dispositivos cumplen con la propiedad de linealidad. Es decir, si la salida del dispositivo representa el filtrado de la suma de dos señales de entrada, el mismo resultado se obtendría de filtrar las dos señales de entrada separadamente y después sumar los resultados.

Al analizar la respuesta al impulso de los filtros lineales vemos que podemos dividirlos en dos grandes grupos, los filtros que tienen una respuesta infinita al impulso (Infinite Impulse Response *IIR*) y los que tienen una respuesta finita (Finite Impulse Response *FIR*), nos

centraremos en estos últimos que son los que utilizaremos. Existen diferencias bastante notables entre estos dos grupos,

IIR



☐ Eficiencia

☐ **Rapidez computacional**

☐ Pueden ser inestables

FIR



☐ Facilidad del diseño

☐ Mayor carga computacional

☐ **Siempre son estables**

En la comparativa se pueden apreciar los motivos por los cuales se ha elegido un filtro de respuesta finita, que es su estabilidad, a pesar de que se necesita más procesador para realizar los cálculos, ya contamos con un microcontrolador que nos provee de rapidez de procesamiento suficiente.

2.3.1 Filtro de respuesta infinita al impulso (IIR)

Los filtros IIR, a diferencia de los FIR, son recurrentes, dependen de los valores de salida anteriores. Como su nombre indica tienen respuesta infinita por lo que, al estimularlos con un impulso, la salida no volverá al reposo.

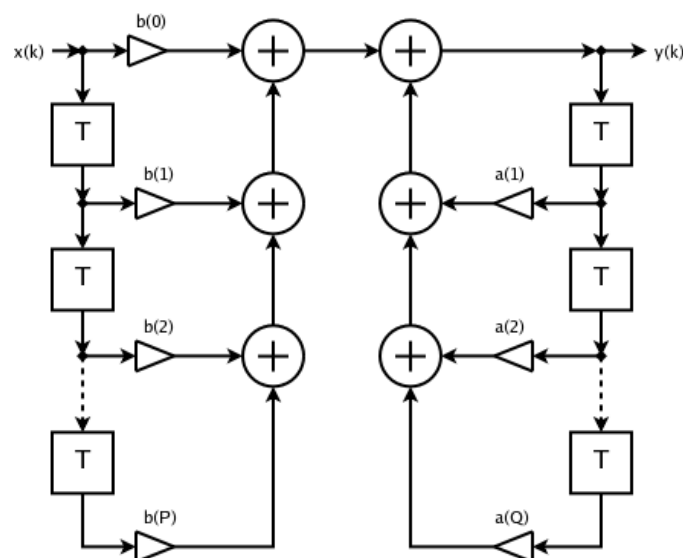


Figura 5 Estructura de un filtro IIR

La salida se calcula con la suma de las entradas y salidas actuales más la suma de salidas anteriores, quedando caracterizado el filtro por la siguiente función de transferencia,

$$H(z) = \frac{\sum_{k=0}^N b_k z^{-k}}{1 + \sum_{k=1}^M a_k z^{-k}}$$

Figura 6 Ecuación IIR

En aplicaciones prácticas la respuesta al impulso, siendo estable, se aproxima a cero en un número finito de muestras.

Existen diferentes tipos de filtros IIR:

- Analógicos (digitalizados)
 - Filtros Butterworth.
 - Filtros Chevyshev.
 - Chevyshev II o filtros inversos de Chevyshev.
 - Filtros elípticos o Cauet.
- Digitales
 - Aproximación de Padé.
 - Aproximación de mínimos cuadrados.

2.3.2 Filtro de respuesta finita al impulso (FIR)

Un filtro FIR obtiene su salida a partir de la entrada actual y de las entradas anteriores, siendo la salida un número finito de términos no nulos. La respuesta a un impulso de entrada siempre es un número finito de muestras. La estructura básica de un FIR es,

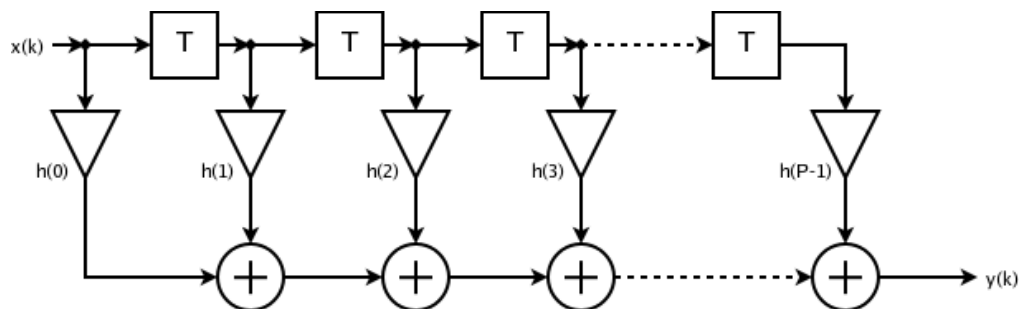


Figura 7 Estructura de un filtro FIR

Pueden hacerse multitud de variaciones de esta estructura según sea más apropiado para su aplicación. La secuencia $y(n)$ de muestras de la respuesta a una secuencia de entrada $x(n)$ puede representarse con la siguiente ecuación:

$$y_n = \sum_{k=0}^{N-1} b_k x_{n-k}$$

Figura 8 Ecuación FIR

Los coeficientes b de la ecuación son la secuencia de muestras de salida del filtro cuando a la entrada hay un impulso.

Al ser estables los filtros FIR se puede elegir un orden tan elevado como se desee, siempre que se tenga en cuenta que, a mayor orden, mayor coste computacional y mayor espacio de memoria utilizado.

2.3.3 Filtro adaptativo basado en el combinador lineal

El combinador lineal adaptativo es una estructura recursiva puesto que hay realimentación de la señal de salida $y(n)$ hacia la entrada $x(n)$. Esta estructura se basa en un filtro FIR y la adaptación de los coeficientes de este filtro, que es el efecto recursivo, se realiza a partir de una señal de error $\varepsilon(n)$ resultante de la diferencia entre la señal de salida y una señal deseada $d(n)$ usada a modo de referencia.

El algoritmo utiliza el error para actualizar los coeficientes, que es lo que nos permite adaptar el filtro en cada iteración y su eficiencia depende de la estructura del filtro y del algoritmo utilizado.

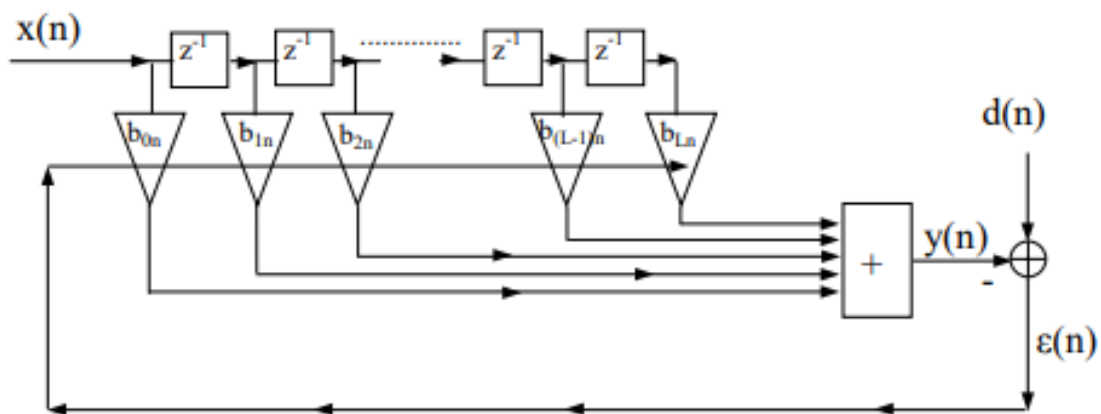


Figura 9 Combinador lineal adaptativo

En la figura 9 se pueden apreciar los elementos de un filtro adaptativo, en primer lugar, tenemos los elementos de retardo y la señal de entrada.



Figura 10 Elementos de retardo CLA

Cada elemento de retardo añadido al filtro aumenta su orden, que implica un mayor coste de computación, pero a cambio el filtro adquiere mayor robustez y velocidad de convergencia. Los elementos de retardo están afectados, cada uno, por un coeficiente del filtro, los cuales se pueden ajustar individualmente.

De esta forma una constante b_{Lk} es multiplicada por la señal del filtro retardada $x(n-k)$ veces, debido a las propiedades de la transformada Z donde z^{-1} opera en la entrada $x(n)$, la salida resultante es $x(n-1)$; combinando finalmente todas las señales mediante un sumatorio, se obtiene la señal de salida $y(n)$.

2.3.4 Filtro adaptativo de error cuadrático medio mínimo (LMS)

Se llama así porque se utiliza el valor medio del cuadrado de la señal de error para obtener el algoritmo que permite ir variando los coeficientes en cada iteración, de tal manera que este error cuadrático medio sea el más pequeño. Esto se hace mediante el método llamado de la pendiente descendente que indica, en cada iteración, la dirección correcta del vector formado por los nuevos coeficientes partiendo de los coeficientes de la iteración anterior.

El combinador FIR de orden L, se puede representar como un registro de desplazamiento de L elementos o retardos y L+1 coeficientes o pesos ajustables que dará lugar a la señal de salida dada por la expresión,

$$y(n) = b_{0n}x(n) + b_{1n}x(n-1) + b_{2n}x(n-2) + \dots + b_{(L-1)n}x(n-L+1) + b_{Ln}x(n-L)$$

Figura 11 Ecuación de la señal de salida

$$y(n) = \sum_{k=0}^L b_{nk}x(n-k)$$

Figura 12 Ecuación de la señal de salida compacta

su representación en forma vectorial vendría dada por la expresión,

$$y(n) = X_n W_n = W_n^T X_n^T$$

Figura 13 Ecuación de la señal de salida en forma vectorial

siendo,

$$X_n = [x(n), x(n-1), x(n-2), \dots, x(n-L+1), x(n-L)]$$

Figura 14 Vector de la señal de entrada

$$W_n = [b_{0n}, b_{1n}, b_{2n}, \dots, b_{(L-1)n}, b_{Ln}]^T$$

Figura 15 Vector de los coeficientes (T := Traspuesta)

Conceptualmente, las expresiones anteriores representan que el vector formado por una muestra de entrada $x(n)$ junto con las L muestras anteriores de entrada multiplicadas escalarmente por el vector formado por sus respectivos coeficientes, proporciona la muestra de salida $y(n)$.

Por otra parte, la muestra de salida $y(n)$ se compara con la respectiva muestra $d(n)$ de la señal deseada para dar una muestra $\varepsilon(n)$ de la señal error. Esto es,

$$\varepsilon(n) = d(n) - y(n) = d(n) - X_n W_n = d(n) - W_n^T X_n^T$$

Figura 16 Ecuación de la señal de error

Para que no haya ambigüedades con el signo conviene considerar el cuadrado de error.

$$\varepsilon(n)^2 = (d(n) - X_n W_n)(d(n) - W_n^T X_n^T) = d(n)^2 - 2d(n)X_n W_n + W_n^T X_n^T X_n W_n$$

Figura 17 Ecuación del cuadrado de la señal de error

La expresión de la figura 17 representa el error cuadrático de una sola iteración. Es decir, el valor del error producido cada vez que una muestra $x(n)$ entra al combinador. Con esta muestra $x(n)$ el combinador calcula el valor de salida $y(n)$ con el conjunto de coeficientes W_n y las muestras anteriores de entrada, entonces el valor $y(n)$ se resta de la muestra $d(n)$ deseada.

Derivando parcialmente el error cuadrático con respecto a los coeficientes, igualando a cero y aplicando el método de la pendiente descendente al error cuadrático medio se calcula el vector de coeficientes actual en función del anterior.

La obtención, en cada iteración, de forma recursiva de los coeficientes es.

$$W_{n+1} = W_n + 2\mu\varepsilon_n X_n^T$$

Figura 18 Ecuación para la obtención de los coeficientes

Escrito en forma de vectores columna es,

$$\begin{bmatrix} b_{0(n+1)} \\ b_{1(n+1)} \\ \vdots \\ b_{L(n+1)} \end{bmatrix} = \begin{bmatrix} b_{0n} \\ b_{1n} \\ \vdots \\ b_{Ln} \end{bmatrix} + 2\mu\varepsilon_n \begin{bmatrix} x(n) \\ x(n-1) \\ \vdots \\ x(n-L) \end{bmatrix}$$

Figura 19 Ecuación para la obtención de los coeficientes en forma vectorial

La constante μ regula la rapidez y estabilidad de la adaptación. Para que el sistema sea estable esta constante debe estar dentro del margen de valores siguientes,

$$0 < \mu < \frac{1}{(L+1)E[x_n^2]} = \frac{1}{(L+1)\text{Potencia media de la señal}}$$

Figura 20 Márgenes de estabilidad ($E := \text{Valor medio}$)

2.4 Aplicaciones

Las aplicaciones de un filtro adaptativo son muy numerosas, ya hemos visto al principio los diversos campos a los que se pueden adaptar, y gracias a que pueden trabajar en ambientes desconocidos satisfactoriamente, a pesar de la ausencia de datos estadísticos, su uso es cada vez más extendido.

No vamos a mostrar cada aplicación, no obstante, sí se puede dividir el filtro adaptativo en cuatro tipos de configuración, según el uso que le queramos dar. Como es lógico se tratará con más profundidad el sistema que nos permite eliminar las interferencias.

Para los siguientes diagramas $x(n)$ será la señal de entrada, $y(n)$ la señal de salida, $d(n)$ la señal deseada y $e(n)$ el error.

2.4.1 Identificador

Un filtro adaptativo cuando se usa como identificador nos proporciona un modelo lineal que represente, más fielmente, la planta del sistema. La planta y el filtro tienen la misma entrada y la salida son el grupo de coeficientes que caracterizan el sistema. Los coeficientes serán estáticos si la planta es estática.

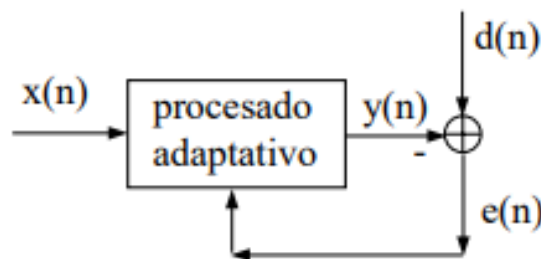


Figura 21 Sistema identificador

2.4.2 Predictor

La función de un sistema predictor es, teniendo suficientes valores anteriores, es predecir la señal deseada. Esta configuración puede ser útil para eliminar ruido aleatorio de una señal periódica, la señal no debe superar la capacidad del filtro para poder conseguir un buen resultado.

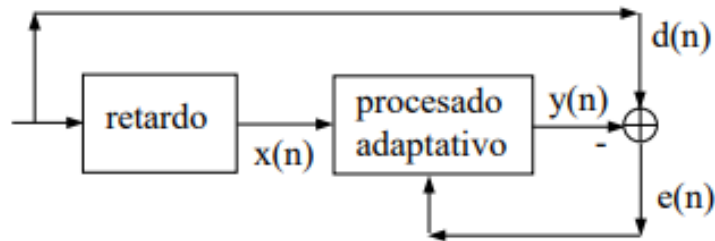


Figura 22 Sistema predictor

2.4.3 Ecualizador-Modelado inverso

Esta configuración nos permite obtener un *modelo inverso* que se adapte a una planta desconocida que este afectada por ruido. Mientras menos se parezcan la señal deseada y la señal de salida, mayor será el error que obtengamos.

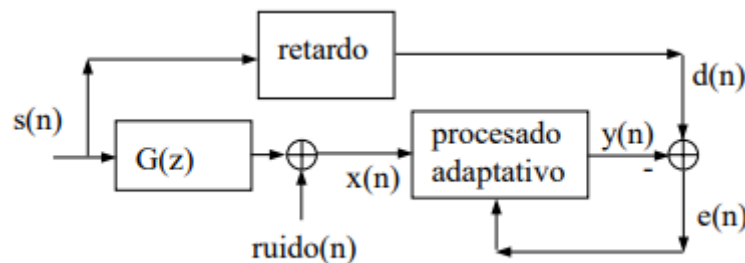


Figura 23 Sistema ecualizador

2.4.4 Cancelador de interferencias

La última clase y en la que se basa este proyecto, es la de un sistema capaz de cancelar las interferencias que tenga una señal. Para poder realizar esta operación necesitamos, a parte de la señal de entrada contaminada con el ruido, una señal que esté correlacionada con la interferencia para poder eliminarla.

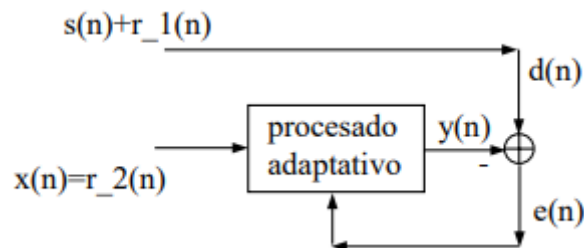


Figura 24 Sistema cancelador de interferencias

En este caso $s(n)+r_1(n)$ es la señal de entrada más la interferencia y $x(n)$ es la señal correlacionada con la interferencia.

Un cancelador de interferencias, como su nombre indica, elimina las interferencias de una señal, aunque por norma general al aplicar un

filtro a una señal elimina parte de la información gracias al filtrado adaptativo se consigue un sistema superior.

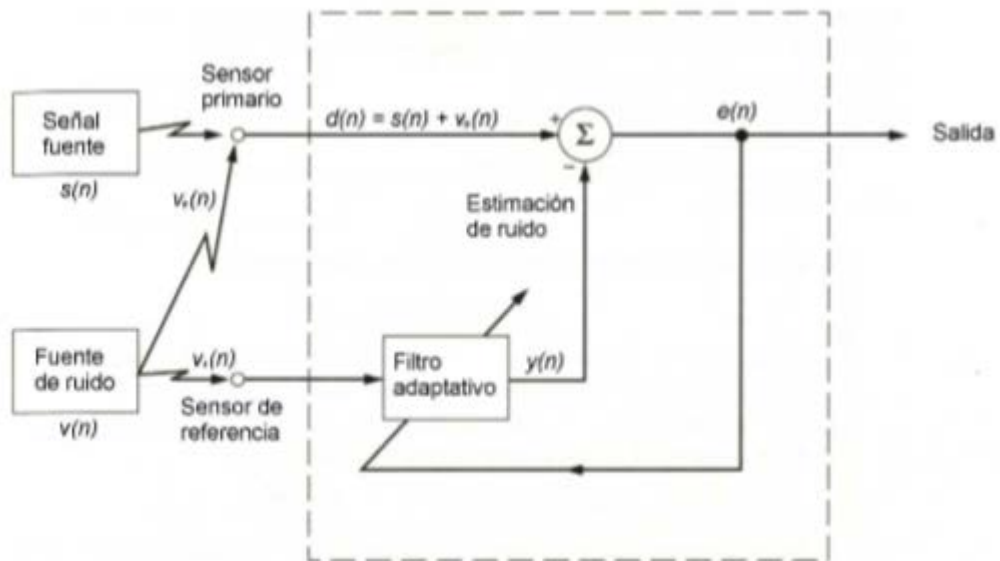


Figura 25 Cancelador de ruido adaptativo

A continuación, se describen diferentes usos que puede tener un cancelador de interferencias:

Cancelación de ruido en señales de voz

Una señal de voz se puede ver seriamente afectada por el ruido ambiental, que es posible eliminarlo utilizando un filtrado adaptativo. En los experimentos realizados por Kang y Fransen en 1987 se colocó un micrófono auxiliar en la cabina de un avión para obtener el ruido ambiental que contaminaba la transmisión de voz de los pilotos, y aplicando un filtro adaptativo basado en el algoritmo de LMS se conseguía una reducción del ruido de entre 10 y 15 decibelios. [1]

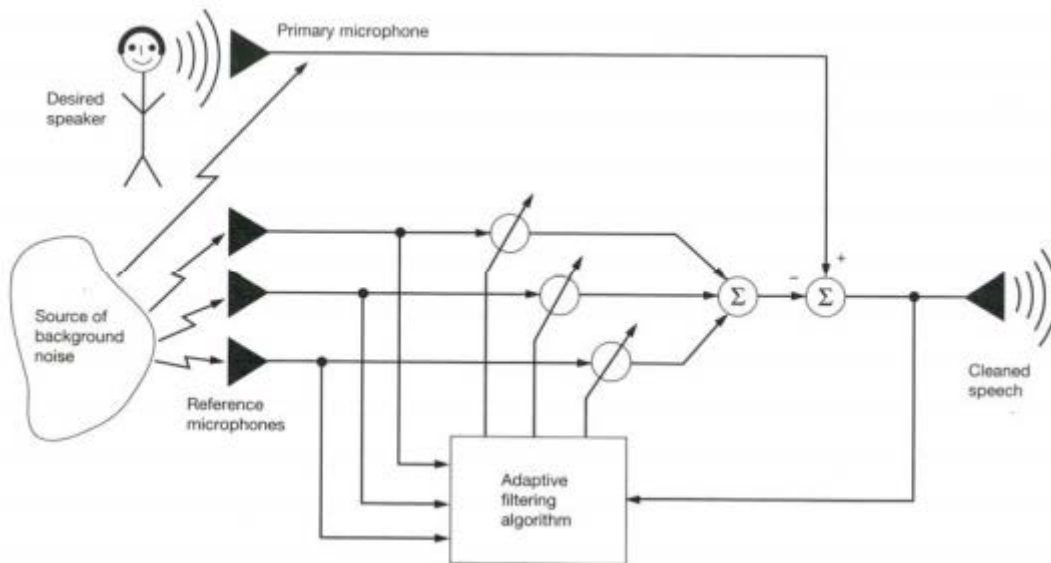


Figura 26 Cancelación adaptativa de ruido

Mejoramiento adaptativo de voz

Utilizando una ampliación del sistema anterior, se puede lograr captar la señal que se desee de una habitación siempre que se distribuyan micrófonos (sondas), por la estancia y a ajustando los coeficientes de los filtros utilizados. De esta forma una persona que tuviera problemas auditivos para poder centrarse en un sonido en concreto podría utilizar un aparato con diferentes receptores que le permitiera seleccionar el rango de audición.

2.5 Conclusiones

En este capítulo, además de describir y clasificar distintos tipos de filtro se han descrito los fundamentos teóricos del filtrado adaptativo LMS basado en el combinador lineal y mostrado el funcionamiento de sus aplicaciones estándar.

Ha quedado patente pues, que utilizar este tipo de sistema nos permite recuperar señales libres de las interferencias que antes contenían. El contenido espectral de las señales así recuperadas presenta un mayor grado de conservación que cuando han sido tratadas mediante filtrado clásico. Gracias a la tecnología actual, es relativamente sencillo y con un bajo coste implementar este tipo de filtrado.

En los próximos capítulos veremos cómo implementaremos la configuración de cancelador de interferencias con en el filtro (LMS), en *Matlab* para su simulación, y en microcontrolador-para comprobar su funcionamiento con el experimento práctico que se realizará.

2.6 Bibliografía

[1] Stergiopoulos, S. (2009). *Advanced Signal Processing: Theory and Implementation for Sonar, Radar, and Non-Invasive Medical Diagnostic Systems. B/W Illustrations.*

Proakis, J. Manolakis, D. (1996). *Digital signal processing. Prentice-Hall.* Cap. 8, 614-692.

Hochman, M. *Dynamical Systems Theory: What in the World is it?*. Einstein Institute of Mathematics. Recuperado el 16 de febrero de 2018 de <http://math.huji.ac.il/~mhochman/research-expo.html>

Nykamp, DQ. *The idea of a dynamical system. Math Insight.* Recuperado el 16 de febrero de 2018 http://mathinsight.org/dynamical_system_idea

Regulador centrífugo. Wikipedia. Recuperado el 5 de junio de 2017 de https://es.wikipedia.org/wiki/Regulador_centro%CC%80fugo

Maxwell, J. (1868). *On Governors*. Proceedings of the Royal Society, No.100.

Luque, F. (2013). *Control adaptativo*. Slideshare. Recuperado el 5 de junio de 2013 <https://es.slideshare.net/frannluque/control-adaptativo-28021034>

Flor Unda, O. (2013). *Control predictivo basado en modelo del campo de colectores solares tipo Acurex, capítulo 3*. Sevilla: Universidad de Sevilla de <http://bibing.us.es/proyectos/abreproy/70451>

Ramírez, J. (2011). *Ruido eléctrico e interferencias*. Universidad de Bogotá <https://ruido.wikispaces.com/RUIDO+EN+COMUNICACIONES>

Lleida, E. (2013). *Filtrado adaptativo*. Universidad de Zaragoza. Recuperado el 15 de marzo de 2018 de <http://physionet.cps.unizar.es/~eduardo/docencia/tds/librohtml/ada pt1.htm>

Filtros IIR y FIR. Universidad del País Vasco. Recuperado el 10 de abril de 2018 de http://www.ehu.eus/procesadoinsvirtual/T6_filtros%20iir%20y%20fir1.html

Martínez, M. Gómez, L. Serrano, A. J. Vila, J. Gómez, J. (2010). *Diseño de filtros IIR*. Universidad de Valencia. Recuperado el 10 de marzo de 2018 de http://ocw.uv.es/ingenieria-y-arquitectura/filtros-digitales/tema_4_diseno_de_filtros_iir.pdf

Martínez, M. Gómez, L. Serrano, A. J. Vila, J. Gómez, J. (2010). *Diseño de filtros FIR*. Universidad de Valencia. Recuperado el 10 de marzo de 2018 de

http://ocw.uv.es/ingenieria-y-arquitectura/filtros-digitales/tema_3._diseno_de_filtros_fir.pdf

3 Algoritmo LMS

En este capítulo se mostrará de forma simplificada el funcionamiento del algoritmo que utilizará nuestro filtro adaptativo, el cual ya hemos hecho referencia antes, el algoritmo *Least Mean Squares* o LMS.

3.1 Introducción

El algoritmo LMS trata de reproducir un filtro deseado ajustando sus coeficientes que permiten obtener el valor esperado mínimo del cuadrado de la diferencia entre la salida del sistema y la señal deseada, es decir el error de la señal.

Este algoritmo pertenece a la familia de los algoritmos de gradiente estocástico, la ventaja de usar un gradiente estocástico descendiente (*GED*) frente a un gradiente descendiente (*GD*) es que, mientras en el *GD* es necesario recorrer todas las muestras para actualizar una sola vez un parámetro en una iteración, para el *GED* necesitas utilizar una sola muestra para realizar la iteración; por lo que es mucho más rápido usar el *GED*, ya que desde la primera muestra empieza a converger y lo hace mucho más rápido que el *GD*, a pesar de que se obtienen resultados más precisos con este último, en la mayoría de los casos los parámetros obtenidos a partir del *GED* son suficientes porque cuando alcanzan sus valores óptimos se quedan oscilando alrededor de estos puntos.

Debido a su velocidad de procesamiento y su simplicidad el algoritmo LMS ha cobrado protagonismo y se utiliza ampliamente en el campo de procesamiento de señal y de control adaptativo.

3.2 Estructura del algoritmo LMS

El algoritmo LMS comprende dos procesos básicos:

- Un proceso de filtrado, que implica el cálculo de la salida generada por un filtro transversal, y la generación de una estimación del error comparando esta salida con la respuesta deseada.
- Un proceso adaptativo, que realiza el ajuste automático de los coeficientes del filtro de acuerdo con la estimación del error.

Un algoritmo LMS de orden p se puede resumir de la siguiente forma:

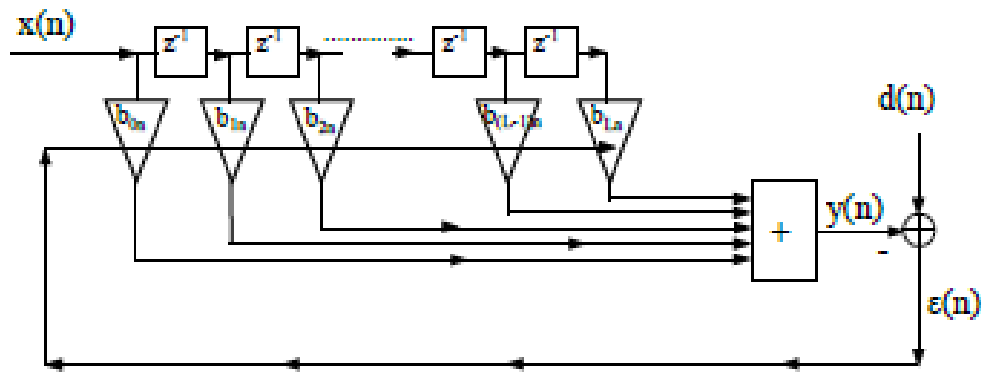
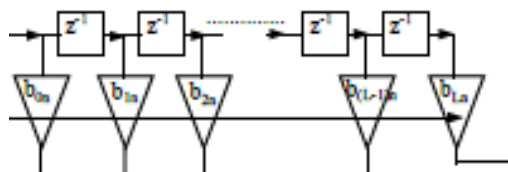


Figura 27 Esquema LMS

Figura 28 Orden del filtro (p)

- Parámetros
 - p = Orden del filtro
 - μ = Paso del filtro
 - n = Numero de muestra actual
 - $X(n)$ = Muestra de entrada
 - $y(n)$ = Muestra de salida
 - $d(n)$ = Señal deseada
 - $\epsilon(n)$ = Error en la señal
 - $W(n)$ = Filtro estimado
- Ecuaciones
 - $y(n) = W(n) \cdot X(n)$
 - $d(n) = y(n) + v(n)$
 - $\epsilon(n) = d(n) - y(n) = d(n) - W(n) \cdot X(n)$
 - $W(n+1) = W(n) + \mu \epsilon(n) \cdot x(n)$
- Computación para $n = 0, 1, 2, \dots$
 - $x(n) = [x(n), x(n-1), \dots, x(n-p+1)]^T$
 - $\epsilon(n) = d(n) - W(n) \cdot x(n)$
 - $W(n+1) = W(n) + \mu \epsilon(n) \cdot x(n)$

El algoritmo nunca conseguirá los valores exactos de los coeficientes y se quedará oscilando entre valores óptimos realizando así una convergencia. Por lo tanto, es necesario elegir un μ adecuado ya que, si es demasiado alto, puede haber una variación grande entre los coeficientes óptimos haciendo que el gradiente pase en cada iteración de positivo a negativo y al revés en la siguiente, creando así una oscilación

no óptima. Si el paso es demasiado pequeño, a pesar de que se obtendrá un resultado óptimo, el tiempo que tardará en converger será excesivamente lento y tendrá un mayor coste computacional.

3.3 Conclusiones

En este capítulo hemos visto, resumidamente, cómo funciona el método numérico LMS, el cual se ha convertido en el algoritmo estándar frente a otros filtros adaptativos lineales, y porqué es el algoritmo que hemos elegido para implementar nuestro eliminador de interferencias.

La relevancia de este algoritmo se debe principalmente a que posee, entre otras, las siguientes propiedades:

- ✓ Robustez y estabilidad.
- ✓ Implementación sencilla tanto en *software* como en *hardware*.
- ✓ Capacidad de operar en entornos desconocidos.
- ✓ Capacidad de rastrear las variaciones de tiempo en la entrada.

A pesar de sus ventajas, hay que tener en cuenta la principal debilidad del LMS, su velocidad de convergencia, por lo que el algoritmo no puede ser utilizado para cualquier aplicación en nuestro caso será más que suficiente, y nos beneficiaremos de sus propiedades.

Debido a su amplia utilización han aparecido variaciones del algoritmo LMS que le permiten adaptarse a una mayor variedad de problemas como por ejemplo el LMS normalizado, el LMS con signo... A pesar de que pueden superar la estabilidad y la robustez del LMS original, su complejidad y su coste computacional también son mayores.

No se ha profundizado en la explicación matemática del algoritmo LMS dado que no es el objeto de este proyecto, en los siguientes capítulos se podrá ver su implementación en *Matlab* y en microcontrolador; de esta forma se ofrecerá una visión práctica del algoritmo y del cancelador de interferencias para que cualquiera pueda replicar la parte experimental y que compruebe por sí mismo la capacidad y utilidad de este sistema.

3.4 Bibliografía

Chow, T. (2000). *Mathematical methods for physicists*. Cambridge University Press. 477-478.

Proakis, J. Manolakis, D. (1996). *Digital signal processing*. Prentice-Hall. Cap. 8, 706-711.

Variants of the LMS algorithm-Lecture 2. Tampere university of technology. Recuperado el 25 de marzo de 2018 de

<http://www.eit.lth.se/fileadmin/eit/courses/ett042/LEC/notes2.pdf>

Adaptive Signal Processing (2011). Lund University. Recuperado el 25 de marzo de 2018 de

<https://www.cs.tut.fi/~tabus/course/ASP/SGN2206LectureNew5.pdf>

Dhiman, J. Ahmad, S. Gulia, K. (2013). *Comparison between Adaptative filter Algorithms. International Journal of Science, Engineering and Technology Research (IJSERT)*. Volume 2, Issue 5, 1100-1103.

Saurabh, R. Bhalchandra, B. (2016). *Optimization of LMS Algorithm for System Identification. International Conference on Signal and Information Processing (IConSIP)*.

Haykin, S. (1998). *Neural Networks: A Comprehensive Foundation. Prentice Hall, 2nd edition. Cap. 5*

4 Experimento práctico

Este capítulo contiene las instrucciones para la realización del experimento, además de los resultados de este. Antes de poder realizar el experimento debemos familiarizarnos con las herramientas que utilizaremos, *Matlab*, un microcontrolador STM32F407 *Discovery* y dos acelerómetros, en este caso ambos analógicos (las fichas técnicas de los dispositivos pueden encontrarse en el anexo), entre otros.

4.1 Elementos necesarios

Para poder realizar el experimento necesitaremos varios tipos de software y hardware, a continuación, se muestra una lista de todos los que se han utilizado en este caso, cualquier software o hardware que pueda realizar las mismas operaciones puede ser utilizado para su realización,

- Hardware
 - 1x Microcontrolador (STM32F407G-DISC1).
 - 2x Sensores para detectar movimiento (Acelerómetro Analógico ADXL 335).
 - 2x Motor de CC.
 - Sistema (Estructura).
- Software
 - Programador para microcontrolador (Atollic TrueSTUDIO).
 - Comunicación y obtención de datos (Tera Term).
 - Tratamiento de los datos (Matlab).

Se mostrará una breve descripción de todos los elementos que se necesitan para la realización del proyecto.

4.1.1 Microcontrolador

Un microcontrolador es un circuito integrado programable, capaz de ejecutar las órdenes guardadas en su memoria [1]. Su funcionalidad se puede dividir en tres partes,

- Unidad Central de Procesamiento (CPU).
- Memoria.
- Elementos periféricos.

Su utilidad reside en estos tres elementos, es capaz procesar diferentes tipos de periféricos y una vez programado únicamente necesita estar conectado a la alimentación. Tienen usos prácticamente ilimitados y se pueden encontrar diseños para aplicaciones muy diferentes. Para este experimento hemos elegido la placa de desarrollo STM32F407G-DISC1 (en adelante la indicaremos como STM), que es de propósito

general e incluye periféricos útiles tales como *clocks*, *ADCs*, *DACs*, *USART*, etc. Y librerías ya creadas para poder trabajar con ellos de forma sencilla.



Figura 29 STM32F407G-DISC1

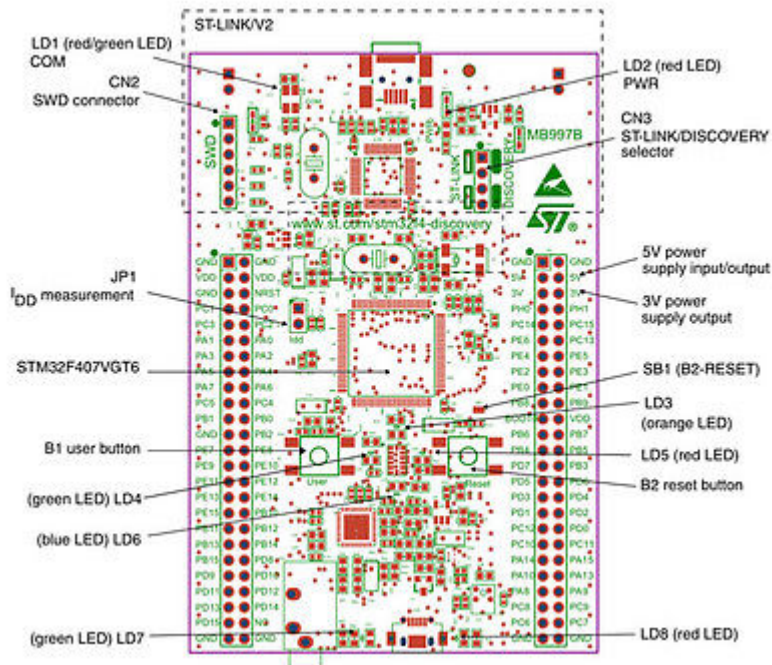


Figura 30 Esquema STM32F407G-DISC1

Por lo tanto, está claro que el elemento principal es el microcontrolador, que se encarga de recoger los datos de los sensores, es necesario que sea capaz de tratar los datos, en nuestro caso al utilizar

acelerómetros analógicos debe ser capaz de transformar y tratar una señal analógica utilizando un ADC (*Analog-to-Digital Conversor*), si se utilizaran señales digitales se debería usar una comunicación I2C, SPI, etc.

El ADC implementado en la STM tiene 12 bits de resolución, es decir, los valores obtenidos tendrán un rango de 0 a 4.095.

4.1.2 Acelerómetros

Un acelerómetro es un dispositivo que, como su nombre indica, nos permite saber la aceleración que se está aplicando sobre él.

En este experimento los acelerómetros tienen un rango de $\pm 3g$ y cuando no tengan aceleración su estado será 0g, el valor mostrado será de 2.047 y sabiendo que cada bit es una variación de 300mV/g podemos obtener la ecuación para mostrar los resultados en mg.

$$a = 1,46484375 \cdot ADC - 2998,53515625$$

Figura 31 Ecuación de los acelerómetros

Para realizar este experimento, trataremos directamente con los valores que nos proporcionará el ADC y los modificaremos para que el rango sea de -2.047 a 2.047.

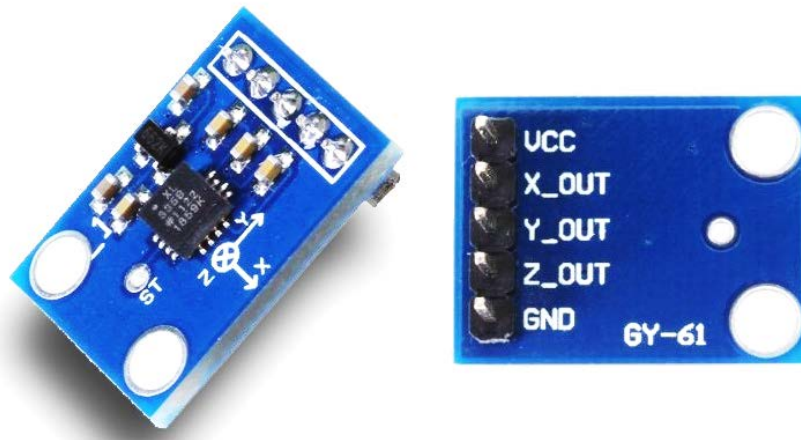


Figura 32 Acelerómetro ADXL 335

4.1.3 Motores de corriente continua

El motor de corriente continua es una máquina que convierte energía eléctrica en mecánica, provocando un movimiento rotatorio, gracias a la acción de un campo magnético. [2]

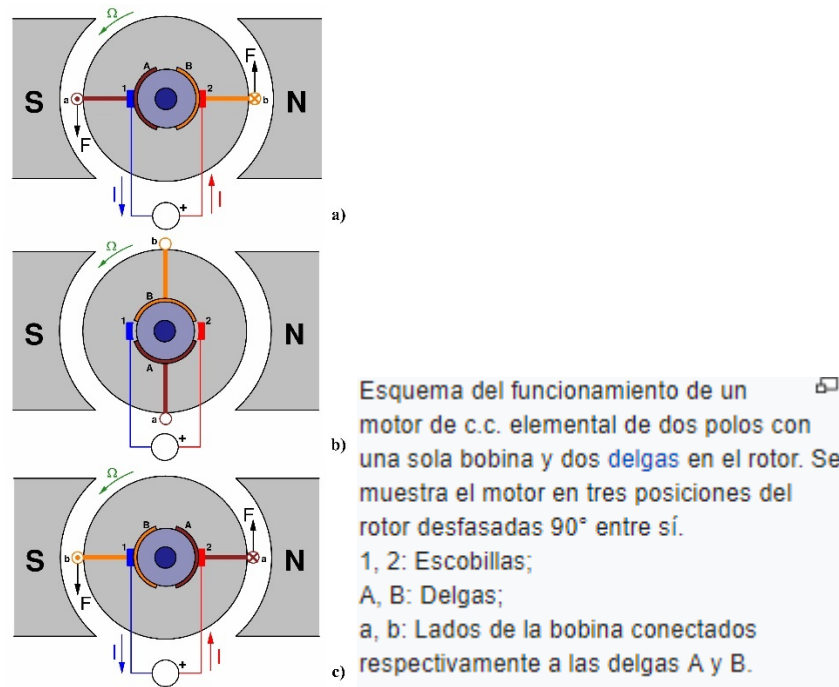


Figura 33 Funcionamiento de un motor de c.c.

Los motores serán los encargados de generar las vibraciones que actuarán sobre el sistema, para que las vibraciones sean más notables se le ha añadido un elemento excéntrico en el eje.

4.1.4 Atollic TrueSTUDIO

Atollic® TrueSTUDIO® para STM32 es un IDE de depuración y desarrollo flexible y extensible para desarrolladores de STM32 MCU, además la compañía de STM lo distribuye libremente. Este software nos permitirá programar nuestro controlador de forma sencilla, como se ha comentado antes, existen muchas librerías ya creadas para la familia de controladores STM32 y este software incluye muchas de estas y en caso de querer añadir otras librerías o crear propias permite administrarlas de forma muy sencilla.

4.1.5 Tera Term

Tera Term nos permitirá enviar y recibir datos del microcontrolador a través de un terminal en el ordenador. Este software es opcional, ya que hay muchas formas de guardar los datos, pero es una de las más sencillas. La conexión se realiza a través de un puerto COM del ordenador donde los datos se envían utilizando el USART del microcontrolador, si el ordenador carece de puerto COM se puede utilizar un USB y emularlo con *software* o *hardware* adicional.

4.1.6 Matlab

MATLAB (MATrix LABoratory) es un entorno informático numérico de múltiples paradigmas. MATLAB permite manipulaciones matriciales,

trazado de funciones y datos, implementación de algoritmos, creación de interfaces de usuario e interconexión con programas escritos en otros lenguajes, incluyendo **C**, **C++**, **C#**, **Java**, **Fortran** y **Python**.

La aplicación *MATLAB* se basa en el lenguaje de *MATLAB scripting*. El uso común de la aplicación implica el uso de la ventana de comandos como un *shell* matemático interactivo o la ejecución de archivos de texto que contienen el código *MATLAB*. Con la instalación de otros paquetes de aplicaciones se pueden realizar operaciones muy heterogéneas, controladores PID, procesamiento de imágenes, control de sistemas, etc. Para este proyecto utilizaremos las herramientas que nos ofrece *MATLAB*, especialmente en filtros digitales.

4.2 Simulación del experimento

Antes de realizar el experimento necesitaremos tener un modo de comprobar que nuestro cancelador de interferencias funciona correctamente, con las herramientas de las que disponemos lo lógico es utilizar *Matlab* para crear una señal y ruido conocidos y filtrarlos, de esta forma comprobaremos que el filtro adaptativo funciona y podremos, de esta forma, comparar los datos del microcontrolador con una simulación en *Matlab*.

Recordemos como es un sistema cancelador de interferencias,

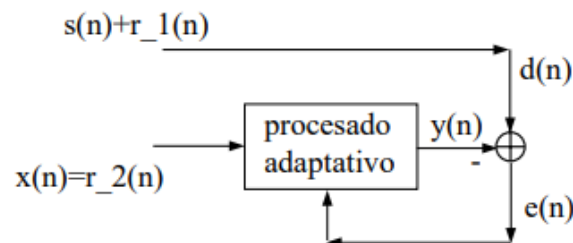


Figura 34 Sistema cancelador de interferencias

Cuando se trata de interferencias en señales cuando se habla del error de la señal, estamos hablando en realidad de la señal que queremos recuperar,

$$\text{Error de la señal} = \text{Señal recuperada}$$

Figura 35 Señal de error en interferencias

Para realizar esta simulación necesitamos una señal conocida, utilizaremos una señal sinusoidal, y le añadiremos el ruido, es importante que el ruido esté correlacionado con la señal, de otra forma no funcionará el filtrado adaptativo. La señal tiene 2500 muestras, amplitud unitaria y la frecuencia es de 1/50 Hz. El ruido generado

aleatoriamente se ha filtrado con un filtro paso bajo para suavizarlo. El código de *Matlab* puede encontrarse en el anexo como *SinAdap.m*.

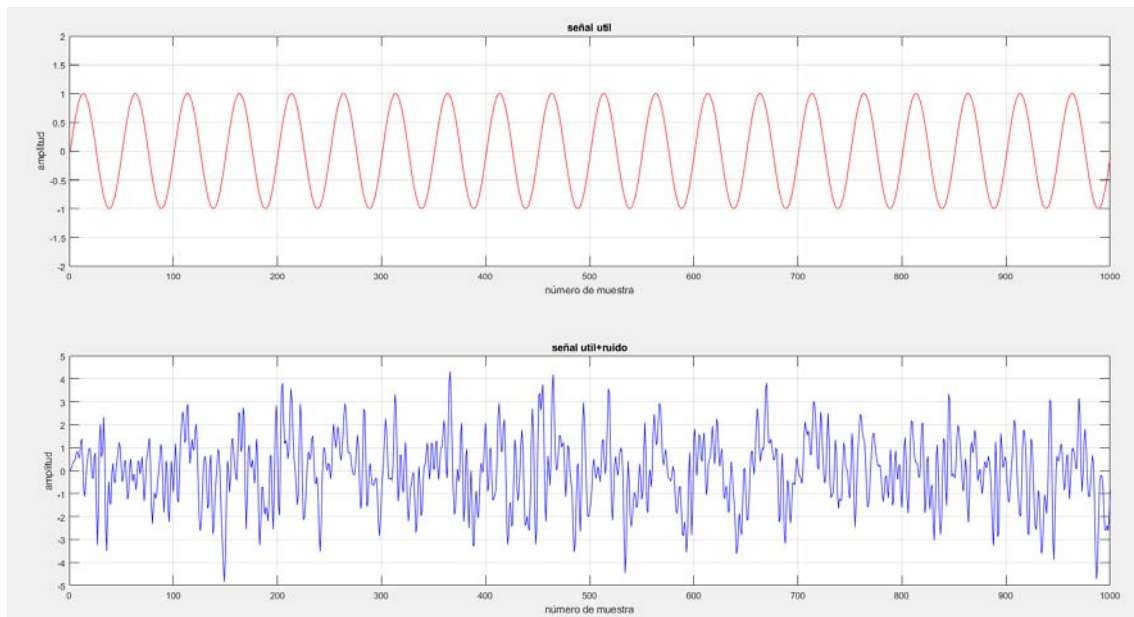


Figura 36 Señal útil y señal útil + ruido

La señal sinusoidal ha quedado distorsionada e irreconocible por el ruido; ahora veremos el ruido correlacionado que le hemos sumado,

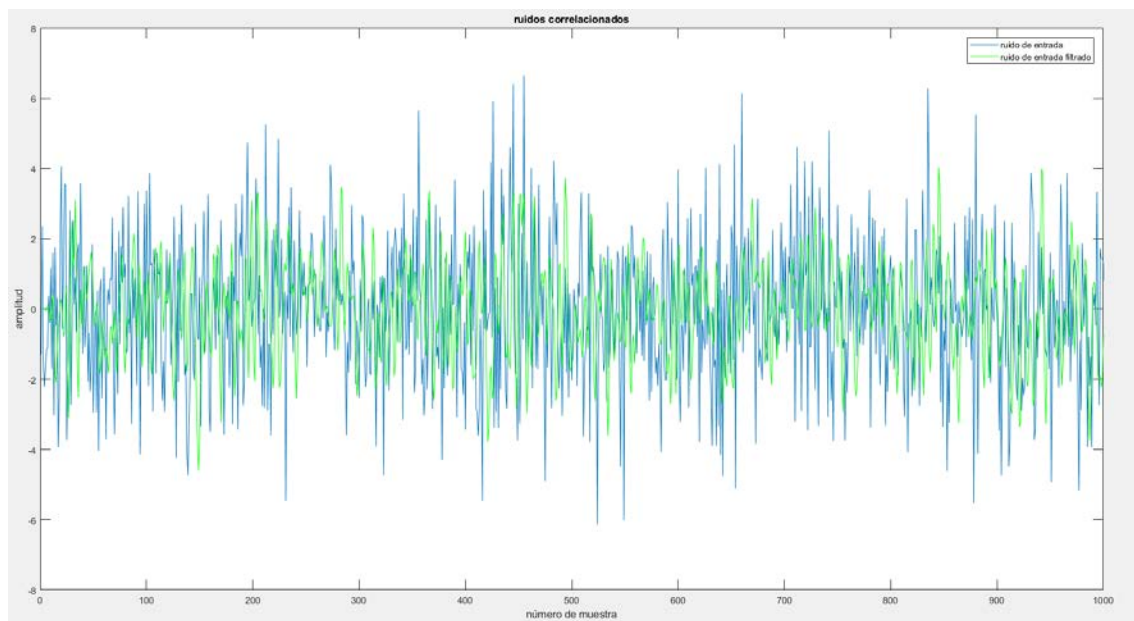


Figura 37 Ruidos Correlacionados

Siguiendo el esquema del sistema se aplica el filtro adaptativo y comprobamos que funciona y que podremos utilizarlo para comparar el filtro que hemos programado en el microcontrolador.

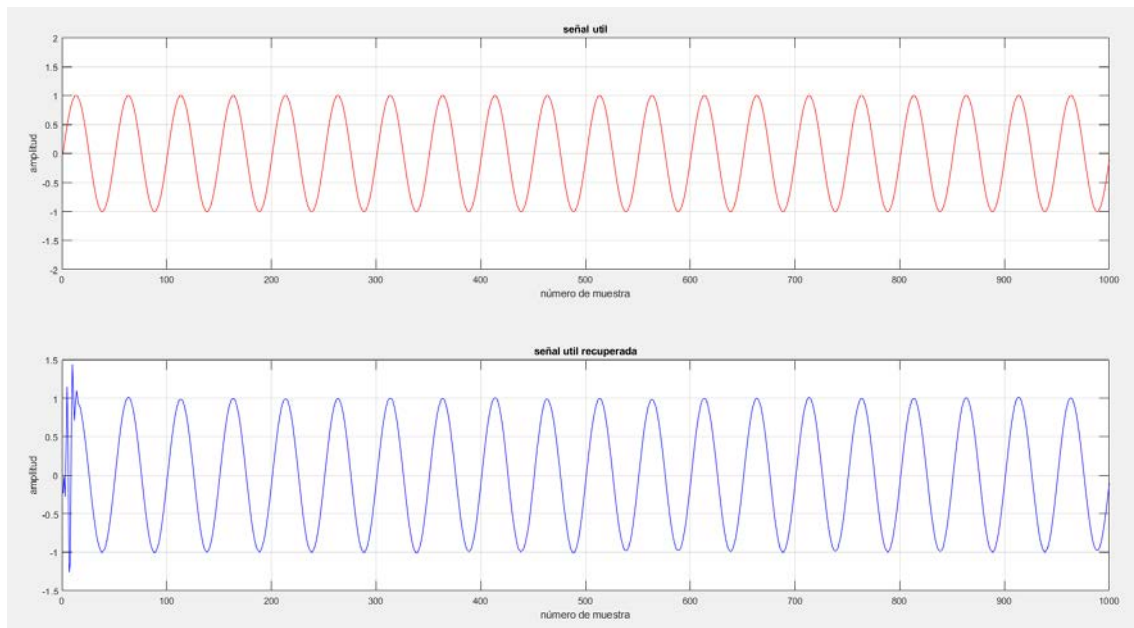


Figura 38 Señal útil y recuperada

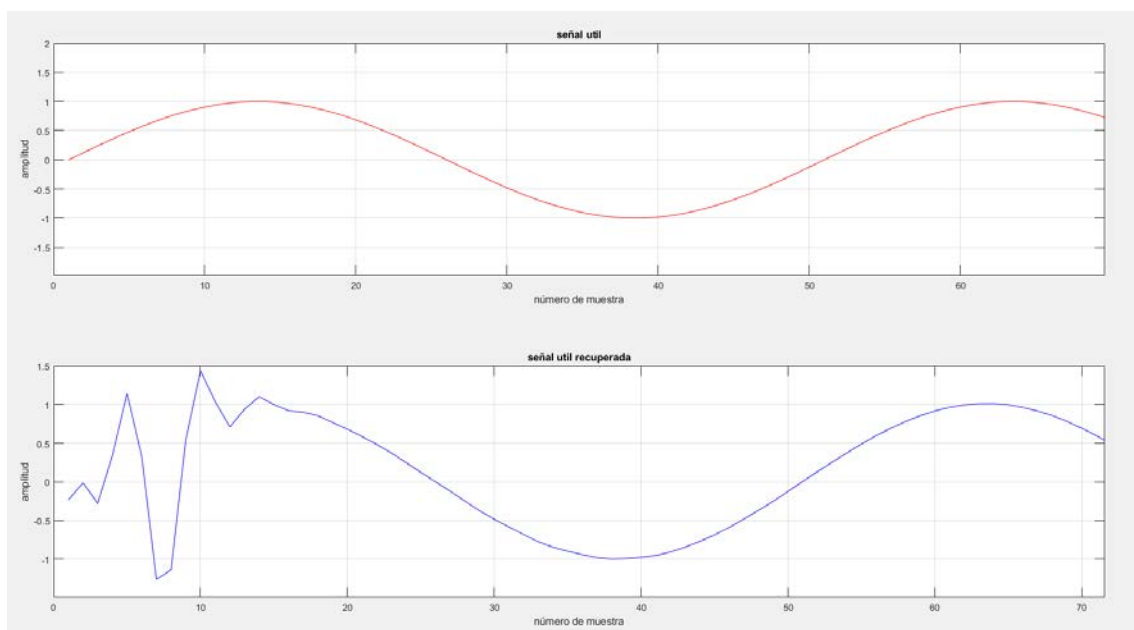
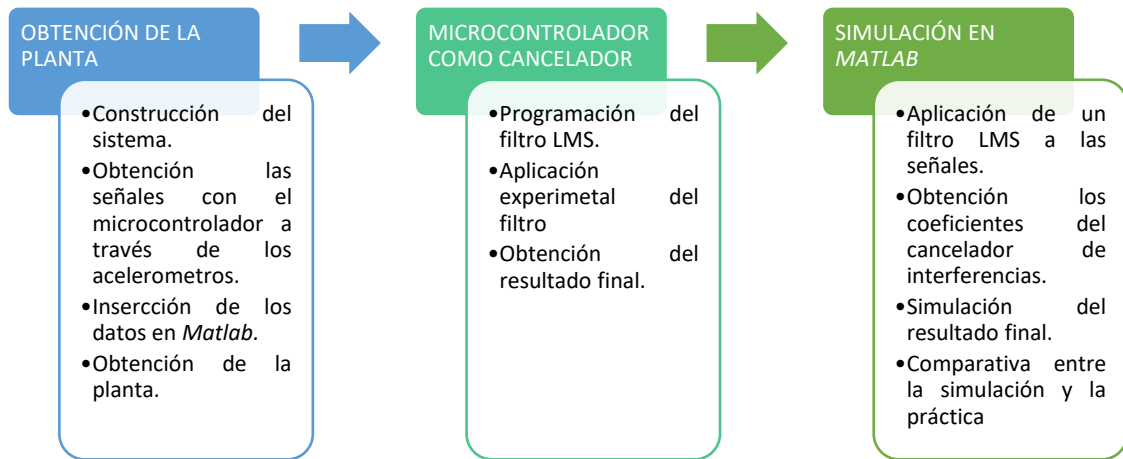


Figura 39 Adaptación del filtro

La señal a quedado prácticamente intacta a excepción de las primeras muestras donde el filtro se va adaptando. El filtro se adapta rápidamente debido a que hemos utilizado 21 coeficientes, es necesario recordar que mientras más coeficientes, mayor es la eficacia del filtro, pero mayor es también su coste computacional.

4.3 Realización del experimento

El primer paso será obtener la planta del sistema sobre el que instalaremos el cancelador de interferencias, una vez obtenida la planta realizaremos una simulación en *Matlab* y a continuación programaremos el microcontrolador con los datos obtenidos y tendremos un cancelador de interferencias operativo.



4.3.1 Obtención de la planta

Como sistema para la realización de todo el experimento se ha construido una estructura de madera que contiene una masa y espuma a modo de amortiguador, para generar las señales deseadas utilizaremos dos motores de corriente continua a los que les hemos añadido un eje excéntrico para aumentar las vibraciones que generan. Los elementos se instalarán en la estructura de la forma que se muestra en las siguientes imágenes.

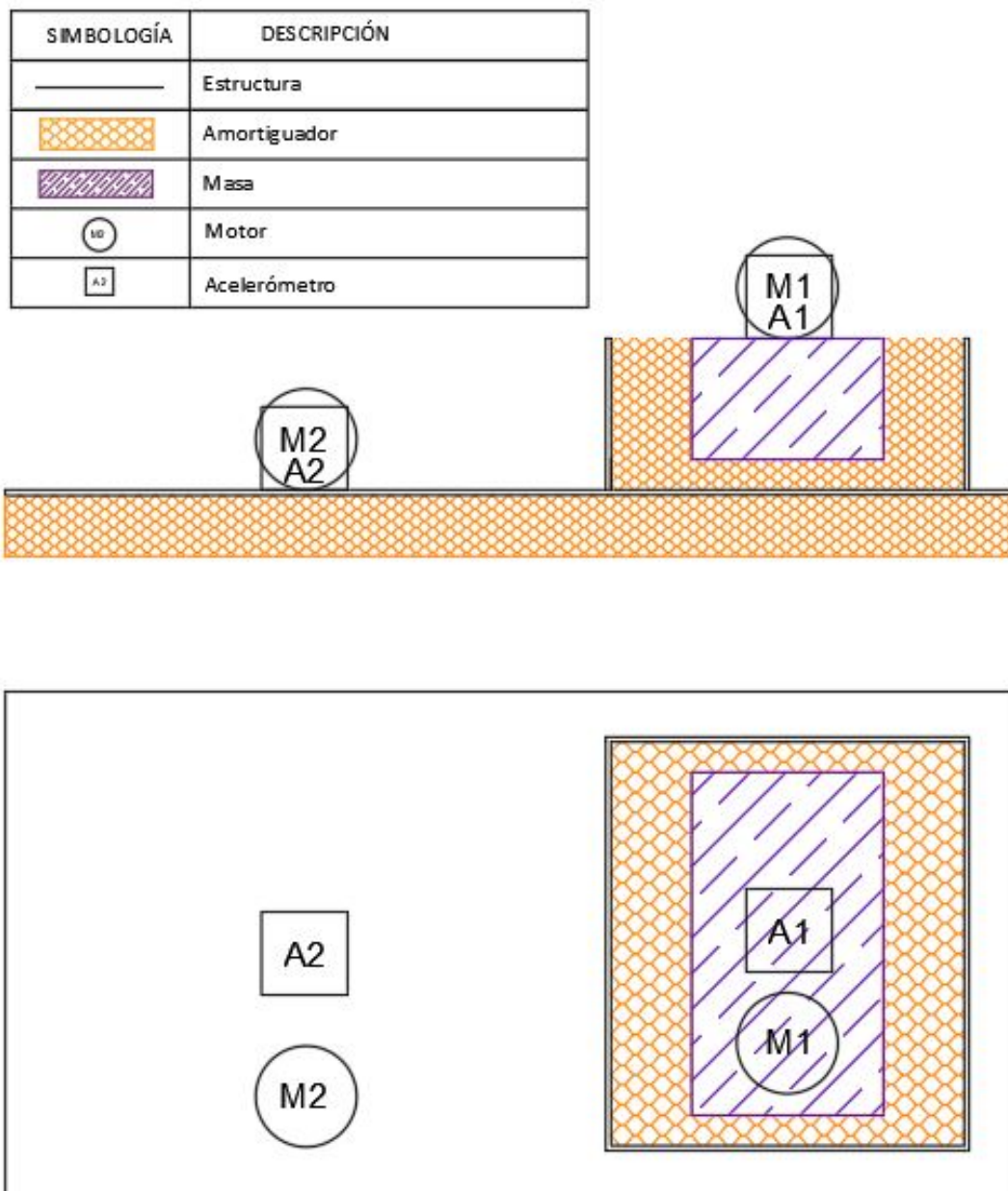
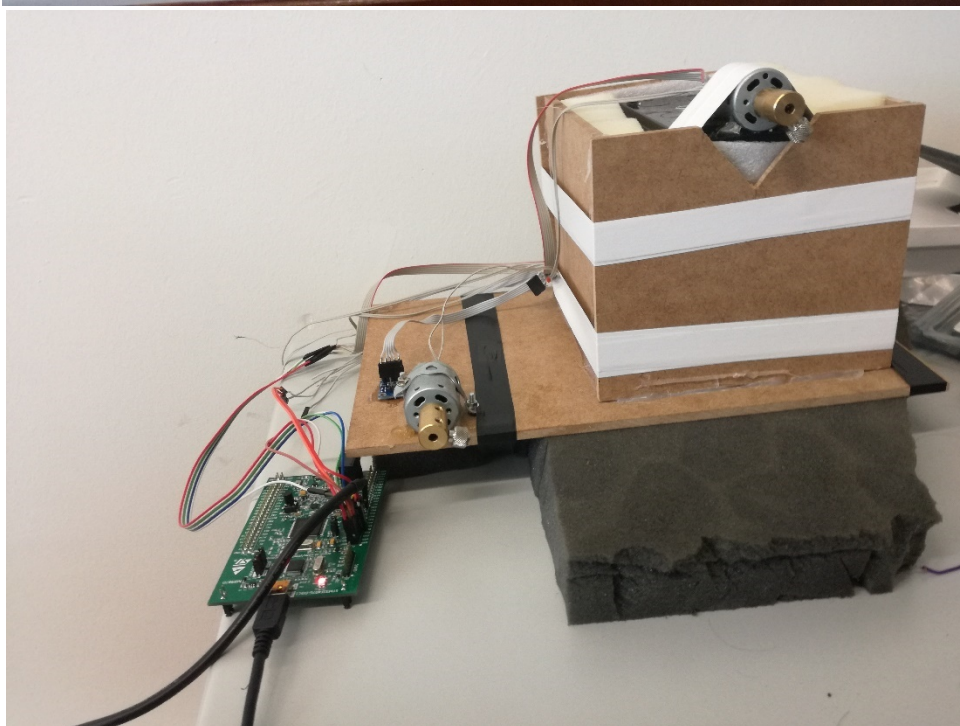


Figura 40 Montaje experimental planos



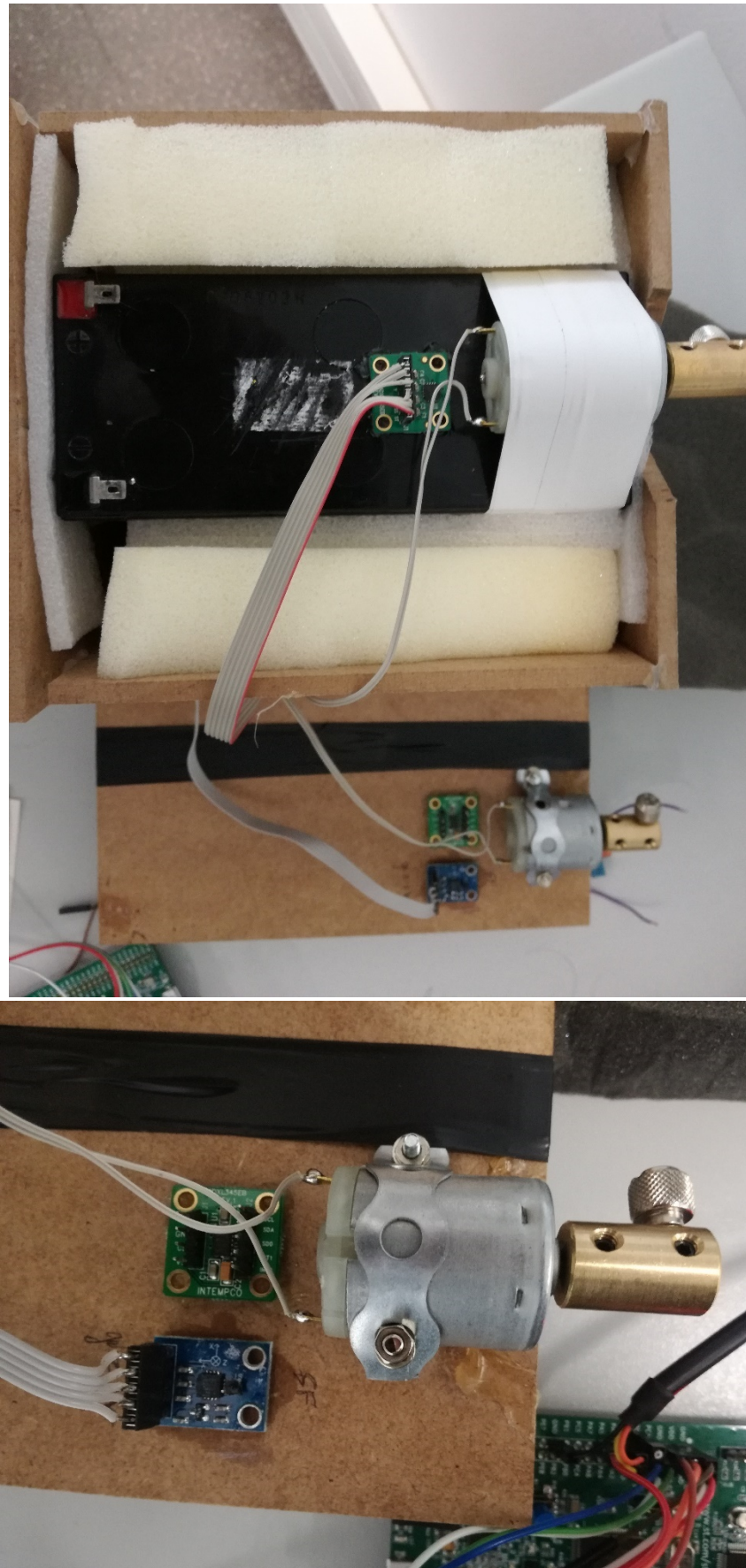


Figura 41 Montaje experimental

Una vez instalados los elementos se procederá a realizar el programa encargado de recoger los datos de los acelerómetros (el código se puede encontrar en el anexo). La elección de acelerómetros analógicos nos facilita la obtención de los datos, ya que únicamente será necesario utilizar los ADC de la STM para realizar la conversión de la señal a un número. El paso final antes de obtener las primeras muestras será realizar las conexiones entre elementos.



Figura 42 Conexiones entre elementos

Para poder guardar las muestras para utilizarlas en *Matlab* utilizaremos *Tera Term*, es necesario configurar de forma correcta la conexión para poder recibir y enviar datos, a través de la terminal se pueden ver los datos transmitidos en tiempo real. Para guardar los datos hace falta activar el *log*.

El primer paso es crear la conexión microcontrolador Tera Term,

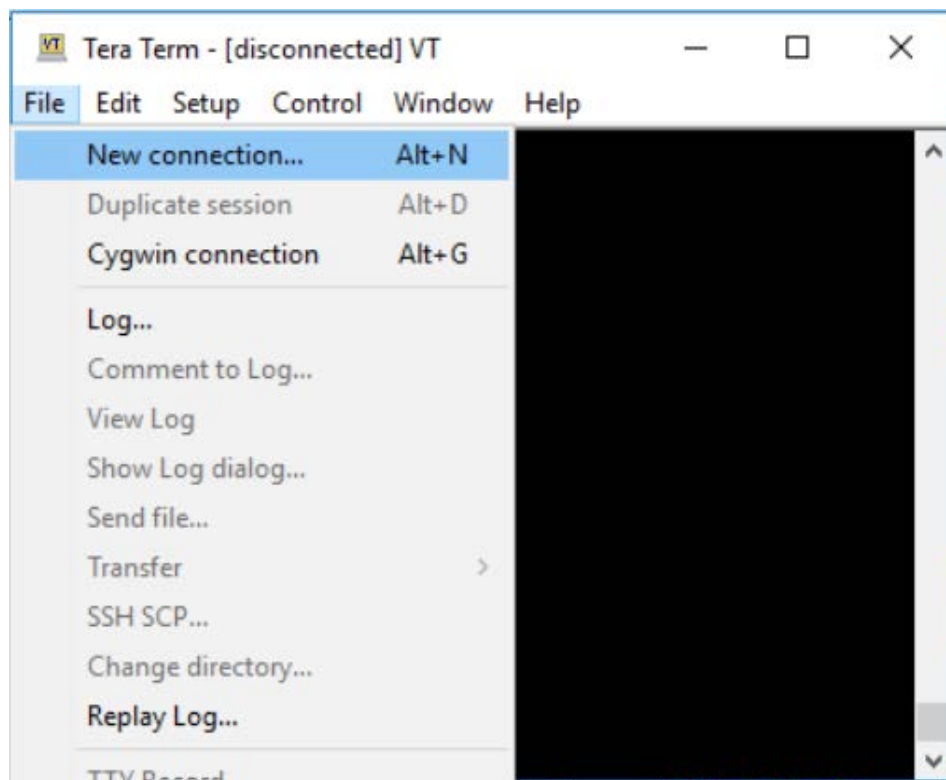


Figura 43 Tera Term - paso 1 – Conexión

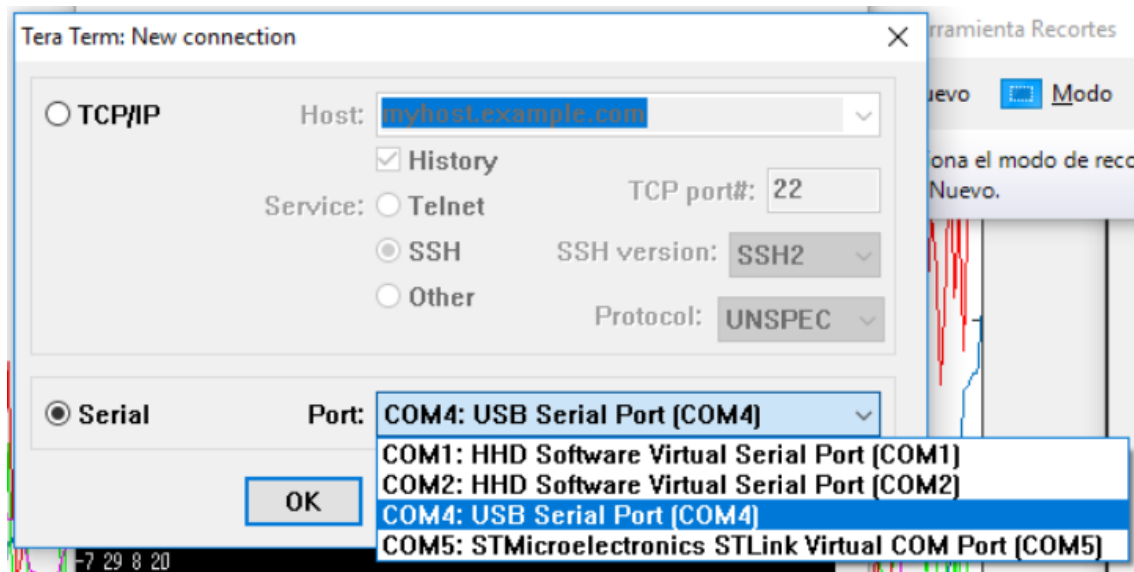
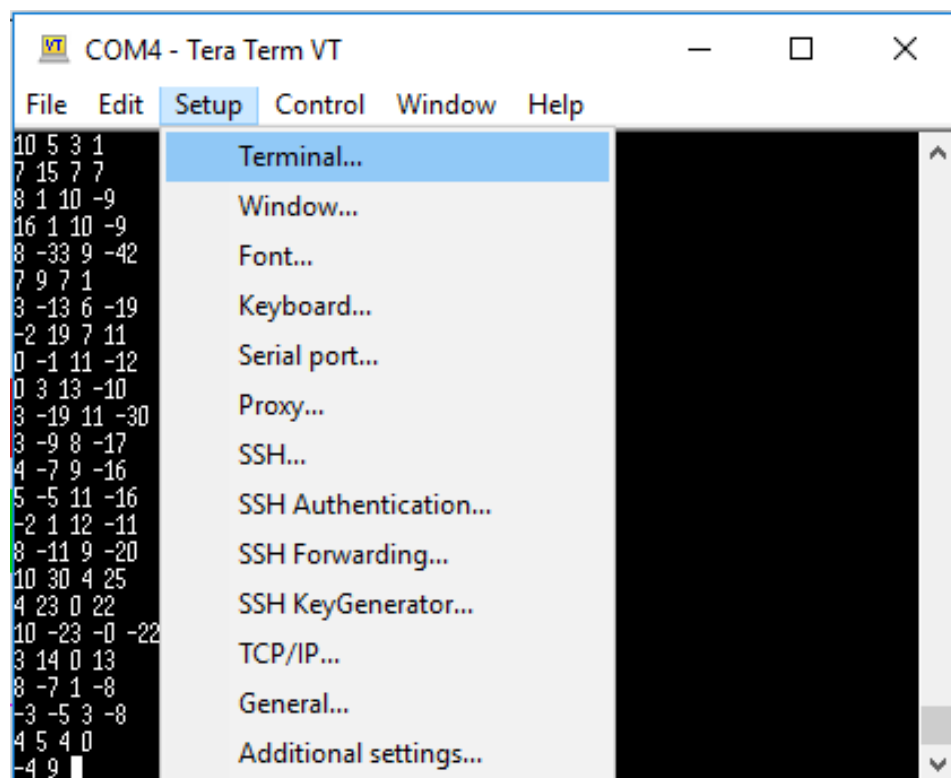


Figura 44 Tera Term - paso 2 - Puerto Serie

El número de puerto COM puede variar según las conexiones que tengas en el ordenador, es importante elegir la conexión *USB Serial Port* y no confundirla con *STLink Virtual COM*.

El siguiente paso es configurar el tipo de datos que se envían, para que *Tera Term* pueda descodificarlos correctamente, el tipo de codificación puede cambiarse en la programación del microcontrolador.



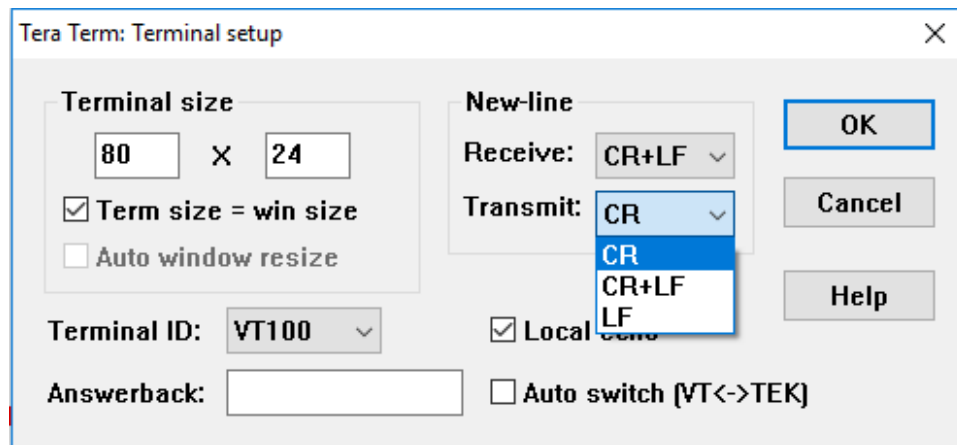
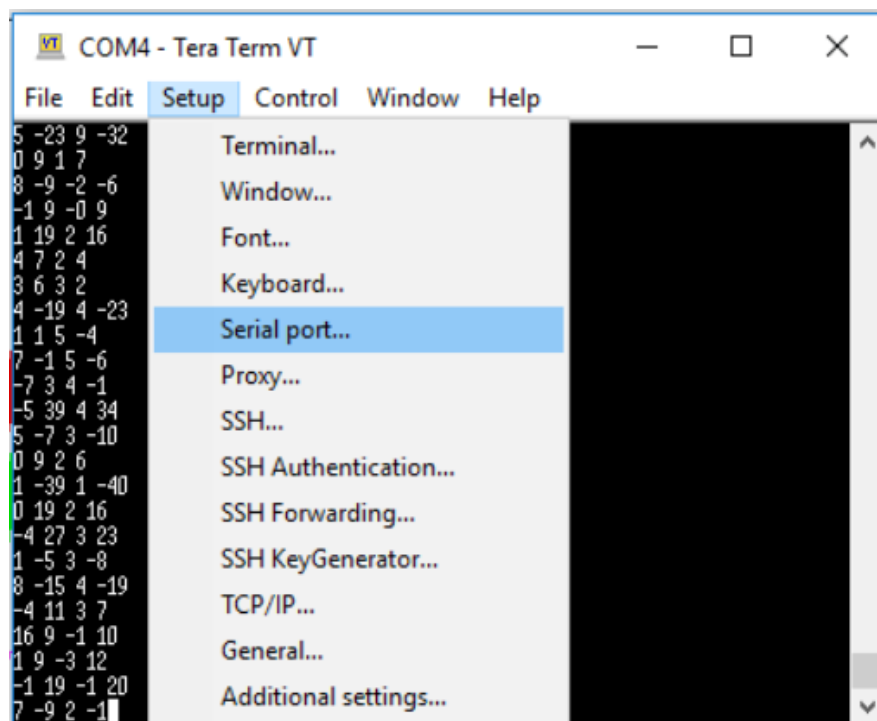


Figura 45 Tera Term - paso 3 - Opciones del terminal



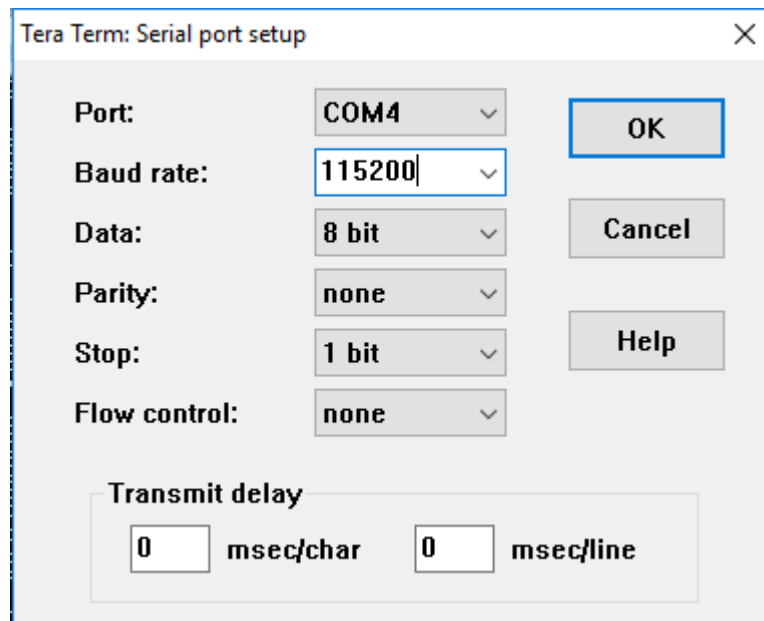


Figura 46 Tera Term - paso 4 - Opciones del puerto serie

Realizada esta configuración deberíamos estar recibiendo cuatro columnas de números, de izquierda a derecha: Interferencia, Señal interferida, Señal de salida y Señal de error.

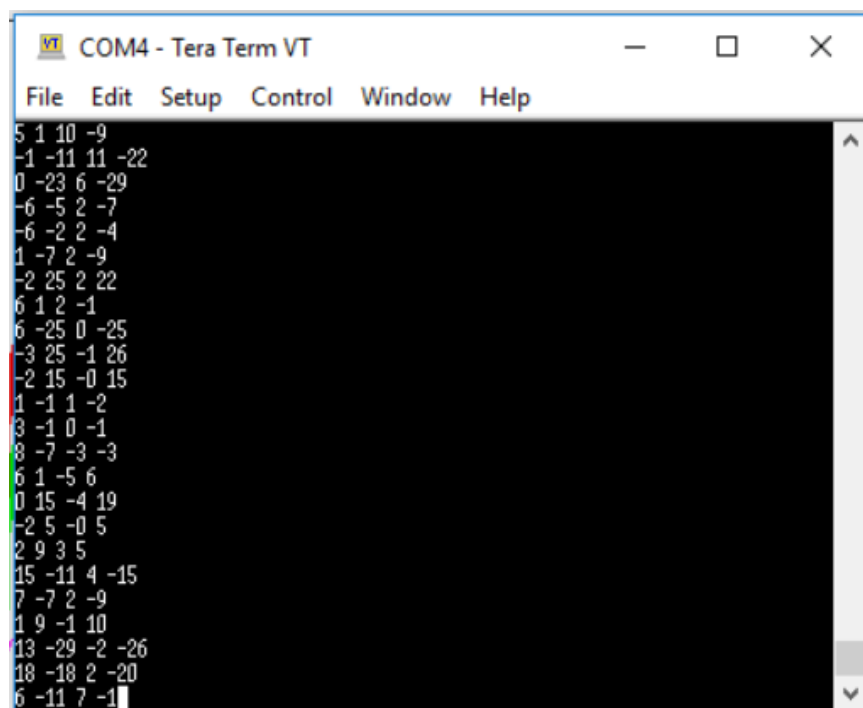


Figura 47 Tera Term - paso 5 - Recibir datos

El paso final es guardar los datos que estamos recibiendo, basta con activar el *log*.

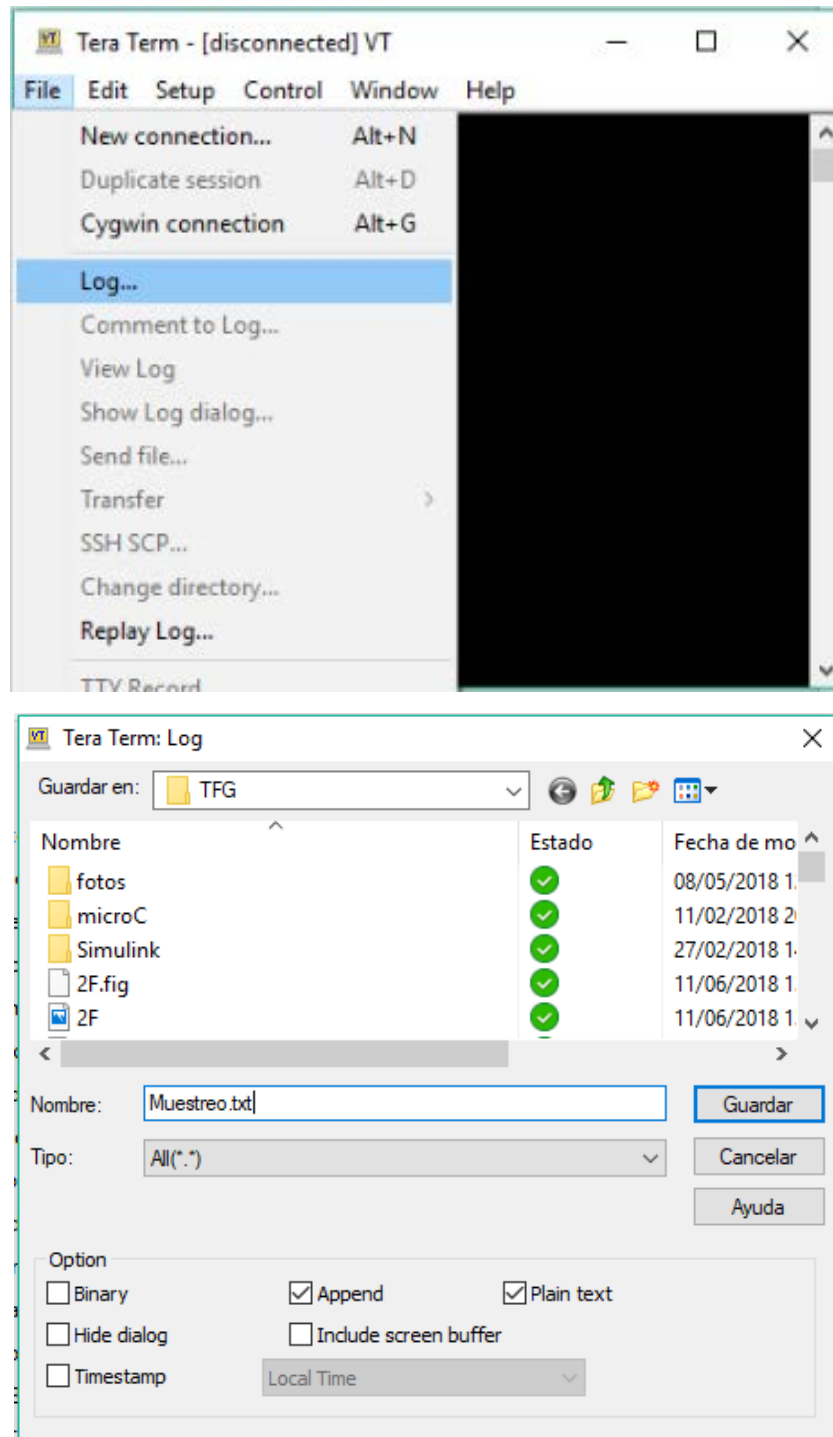


Figura 48 Tera Term - paso 6 – Guardar datos

En nuestro caso le asignaremos la extensión *.txt*, de esta forma, nos resultará muy sencillo importar los datos a *Matlab*. A continuación, se abrirá el dialogo del *log* y podremos pausar o terminar la adquisición de datos.

Para obtener la planta utilizaremos otro uso de los filtros adaptativos que hemos explicado anteriormente, se trata del identificador de sistemas.

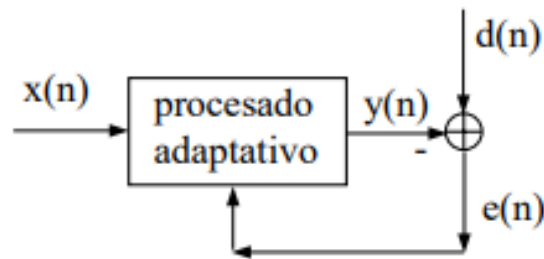


Figura 49 Sistema identificador

Gracias a este procesamiento podremos estimar cómo se comporta nuestra planta y obtener unos coeficientes iniciales para nuestro filtro adaptativo. Para realizar este proceso únicamente debe estar activada la fuente de ruido y adquirir muestras con ambos acelerómetros.



Figura 50 Configuración para la obtención de la planta

Se deben ajustar el número de coeficientes y el salto de ajuste LMS hasta reducir el error a unos niveles aceptables, entonces se pueden utilizar los coeficientes que habrá generado *Matlab*. Una vez obtenidos los coeficientes, podemos realizar la transformada de Fourier y aplicar el valor absoluto, obteniendo de esta forma, el comportamiento del sistema. El código en Matlab para este proceso se puede encontrar en el anexo como *IDSys.m*.

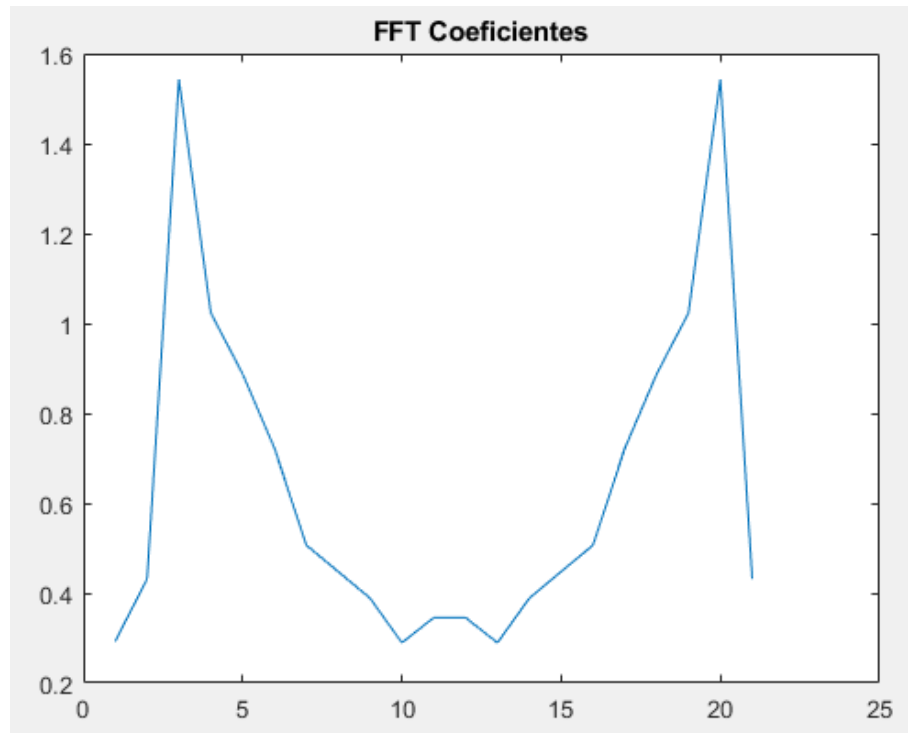


Figura 51 Comportamiento del sistema experimental

Se aprecia de forma clara que nuestro sistema se comporta como un filtro pasa banda, por lo que utilizar unos coeficientes iniciales de un filtro de este tipo dará buenos resultados en el cancelador de interferencias para nuestro sistema.

4.3.2 Microcontrolador como cancelador de interferencias

En este apartado se trata el tema central del proyecto, averiguar si puede usarse un microcontrolador como cancelador de interferencias. Ya se ha explicado cómo realizar las conexiones y como recoger los datos necesarios, únicamente queda programar nuestro cancelador, para eso es importante recordar cómo funciona el combinador lineal adaptativo.

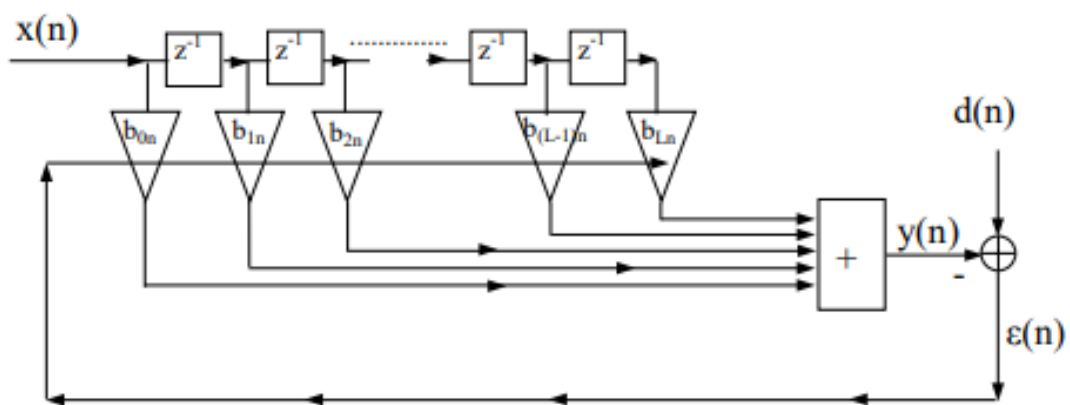


Figura 52 Combinador Lineal Adaptativo

Teniendo la señal interferente ($x(n)$) y la señal interferida ($d(n)$), el primer paso es inicializar dos vectores, uno vacío y otro con los coeficientes del filtro, la longitud de ambos vectores dependerá del orden del filtro.

El proceso es el mismo que se utilizará en *Matlab*, con la excepción de que muchas de las funciones ya existen y por ende simplifica el código, como por ejemplo realizar el producto escalar de dos vectores.

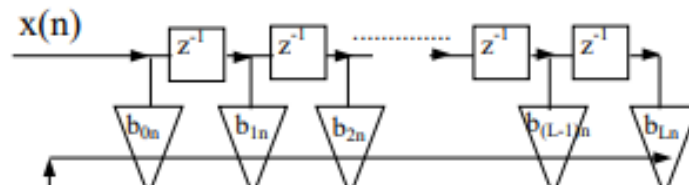


Figura 53 uC - Declaración de vectores

La siguiente operación será desplazar el vector $x(n)$ una muestra hacia la derecha y guardar la nueva muestra en la primera posición del vector. A continuación, se debe realizar el producto escalar del vector de muestras con el vector de los coeficientes, obteniendo así la señal de salida para esa muestra.

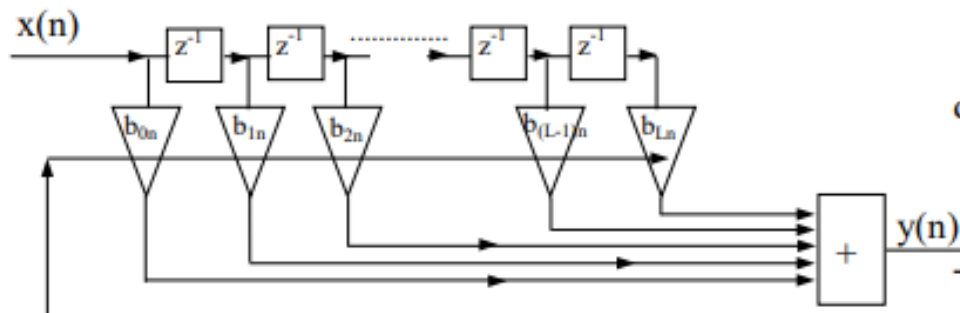


Figura 54 uC - Producto escalar

Seguidamente, se calcula el error y por último se actualizan los coeficientes. Se debe recordar que el error es la señal ya filtrada.

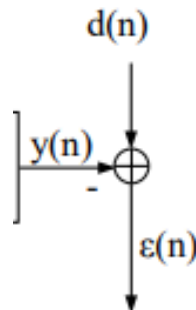


Figura 55 uC - Cálculo del error

La ecuación para el cálculo de los nuevos coeficientes es la siguiente,

$$b(n) = b(n) - \mu \cdot \text{error} \cdot x(n)$$

Figura 56 Ecuación para el cálculo de coeficientes

Donde μ es el salto de ajuste LMS.

Una vez programado se deberá pasar cada muestra que se recoge por el filtro y enviar o guardar los datos de la salida, podría programarse una interfaz que grafique los datos en tiempo real en un monitor. Como el objeto de este proyecto es demostrar que puede funcionar un cancelador de interferencias en un microcontrolador nos bastará con recoger los datos en un documento de texto, tal y como se ha explicado en el apartado anterior.

Si se han realizado todos los pasos correctamente deberíamos obtener unos resultados como los siguientes,

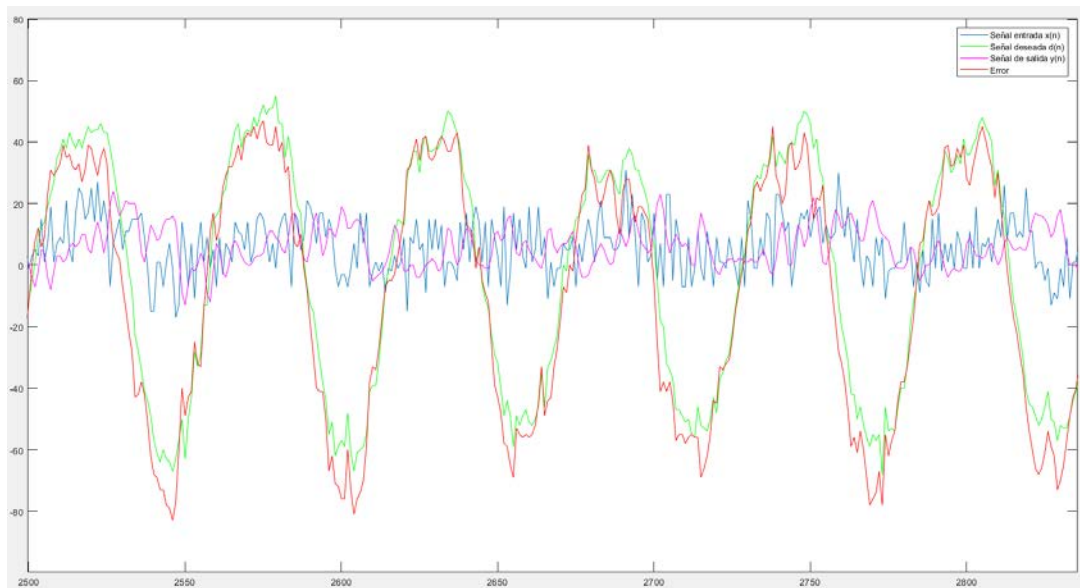


Figura 57 Resultados del cancelador de interferencias en microcontrolador

La señal de error es la señal que obtendríamos si no existiera la interferencia, para asegurarnos de que el filtro ha funcionado correctamente, realizaremos el mismo proceso, pero utilizando *Matlab*. Para mostrar los datos en gráficos se ha utilizado el código *LecturaDatos.m* que se puede encontrar en el anexo.

4.3.3 Simulación en *Matlab*

Para obtener los resultados en *Matlab* utilizaremos el mismo código con el que filtramos la señal sinusoidal en el apartado 4.2. El cambio por realizar es que no será necesario generar señales, puesto que debemos utilizar las mismas que hemos recogido y tratado con el microcontrolador. El código utilizado es *FiltroAd.m* que se puede encontrar en el anexo.

Al utilizar los datos recogidos hemos obtenido estos resultados.

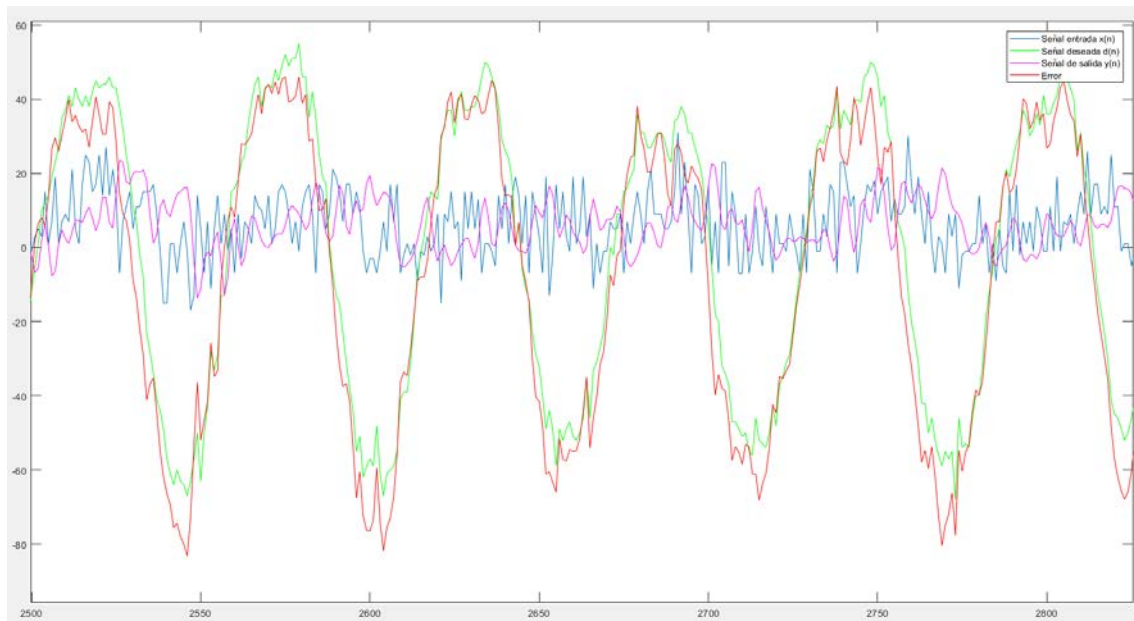


Figura 58 Resultados del cancelador de inferencias en Matlab

Si comparamos las señales de error del microcontrolador con la que acabamos de procesar en *Matlab* podemos determinar si el filtro diseñado para el microcontrolador funciona correctamente.

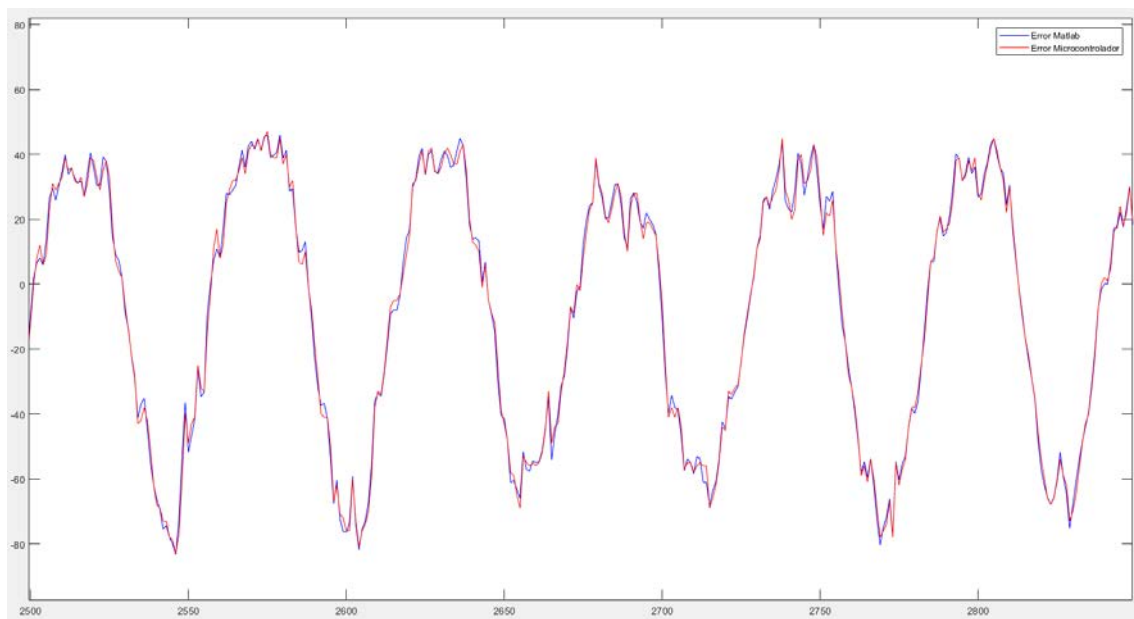


Figura 59 Comparativa de las señales de error

Las señales son prácticamente exactas, por lo tanto, el filtro adaptativo del microcontrolador funciona correctamente y es capaz de cancelar las interferencias con buenos resultados, las pequeñas variaciones se deben a que *Matlab* tiene una mayor precisión que el microcontrolador cuando realiza los cálculos.

4.4 Bibliografía

[1] *Microcontrolador* (2011). Wikipedia. Recuperado el 16 de mayo de 2018 de <https://es.wikipedia.org/wiki/Microcontrolador>

[2] *Principios de funcionamiento del motor de corriente continua*. Automatismoinustrial. Recuperado el 16 de mayo de 2018 de <https://automatismoinustrial.com/motores/1-3-5-motores-de-corriente-continua/1-3-5-2-principios-de-funcionamiento/>

Tera Term. OSDN. Recuperado el 20 de mayo de 2018 de <https://osdn.net/projects/ttssh2/>

MATLAB. Mathworks. Recuperado el 20 de mayo de 2018 de <https://es.mathworks.com/products/matlab.html>

Adaptive Filters. Mathworks. Recuperado el 20 de mayo de 2018 de <https://es.mathworks.com/help/dsp/adaptive-filters.html>

Clariá, F. *Identificación de sistemas control adaptativo*. Revisión 2017. Escuela Politécnica Superior – Universidad de Lérida. Cap. 6.

5 Conclusiones

Tras la realización del experimento práctico ha quedado patente que un microcontrolador es capaz de realizar la función de cancelador de interferencias. Este hecho junto a la capacidad de tratar entradas y salidas y la aparición de nuevos y más potentes microcontroladores demuestra el potencial que tienen en los campos industriales y tecnológicos. Es posible que en el futuro los PLC que actualmente se utilizan para la automatización de la mayor parte de la industria sean substituidos por microcontroladores, ya que son más potentes y baratos, aunque por el momento no tienen el mismo nivel de estandarización y su programación es más complicada y requiere una extensa comprensión del lenguaje C y de las librerías que se deben usar.

Podemos afirmar que el experimento ha logrado con su objetivo y que tanto el microcontrolador, el filtro adaptativo y el algoritmo LMS han cumplido sus funciones correctamente. El filtro ha sido capaz de adaptarse a la señal rápidamente y eliminar la interferencia, es importante elegir unos coeficientes iniciales que al menos se asemejen a un filtro simple, porque si se dejan vacíos el filtro tarda demasiado en adaptarse y en estabilizarse.

En resumen:

- **Correcto funcionamiento de los elementos del experimento, el microcontrolador posee capacidad suficiente para utilizarse como cancelador de interferencias, consiguiendo resultados satisfactorios.**
- Es necesario que la señal interferente este correlacionada con la señal interferida para un funcionamiento correcto del filtro.
- La mala elección de los coeficientes iniciales puede provocar que el filtro se vuelva inestable.
- La complejidad de la programación del microcontrolador, en caso de que no exista una librería específica para la función que se desea realizar, puede suponer un aumento considerable de la carga de trabajo, especialmente si se trata de la programación de protocolos de comunicación, I2C, SPI, USART, etc.
- La creación de librerías estandarizadas para microcontroladores podría suponer un avance hacia su utilización en la industria.

5.1 Coste del estudio

Concepto	Tiempo (h)
Investigación teórica	55
Planificación y construcción del sistema experimental	25
Realización de los experimentos	70
Redacción documental	30
TOTAL	180

Concepto	Unidades	Coste Ud. (€)	Coste (€)
Estructura de madera	1	6	6
Motor CC	2	14	28
Elemento excéntrico	2	4,50	9
Acelerómetro	2	20	40
Amortiguador	1	5	5
STM32F4DISCOVERY	1	17	17
TOTAL (€)	-	-	105

5.1.1 Elementos del proyecto

Motor CC

https://www.amazon.es/gp/product/B00R1LK4CA/ref=oh_aui_detailpage_o01_s00?ie=UTF8&psc=1

Elemento excéntrico

https://www.amazon.es/gp/product/B00DCAJ4NE/ref=oh_aui_detailpage_o00_s00?ie=UTF8&psc=1

Acelerómetro

<http://www.cetronic.es/sqlcommerce/disenos/plantilla1/seccion/producto/DetalleProducto.jsp?idIdioma=&idTienda=93&codProducto=999334053&cPath=1343>

STM32F4DISCOVERY

<https://www.mouser.es/ProductDetail/STMicroelectronics/STM32F407G-DISC1?qs=sGAEpiMZZMtw0nEwywcFgCOvL%2fCIMT%2f2vfdud10VWB0Pm3RHtD6n9g%3d%3d>